

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**POROVNÁNÍ ALGORITMŮ PRO SYNTETICKÉ  
MODELOVÁNÍ DOZVUKOVÝCH VLASTNOSTÍ MÍSTNOSTI**

COMPARISON OF ALGORITHMS FOR CREATION OF ARTIFICIAL ROOM REVERBERATIONS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Tomáš Beňo**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Petr Frenštátský**

**BRNO 2017**



# Bakalářská práce

bakalářský studijní obor **Audio inženýrství**  
Ústav telekomunikací

**Student:** Tomáš Beňo

**ID:** 174443

**Ročník:** 3

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## Porovnání algoritmů pro syntetické modelování dozvukových vlastností místnosti

### POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je seznámení se s klasickými algoritmy pro modelování dozvukových vlastností, jako jsou Datorrův, Gardnerův a Jotův algoritmus. Student provede porovnání z hlediska možnosti řízení dozvukových vlastností a komplexnosti algoritmů. Dané algoritmy budou implementovány v prostředí Matlab.

### DOPORUČENÁ LITERATURA:

[1] SYSEL, P.; SMÉKAL, Z. Číslicové filtry. Brno: Vysoké učení technické v Brně, 2012. s 1-148. ISBN 978-8-214-4454-6

[2] EDITED BY MARK KAHRS, Karlheinz Brandenburg. Applications of digital signal processing to audio and acoustics. New York: Kluwer Academic, 2002. ISBN 9780306470424.

**Termín zadání:** 1.2.2017

**Termín odevzdání:** 8.6.2017

**Vedoucí práce:** Ing. Petr Frenštátský

**Konzultant:**

**doc. Ing. Jiří Mišurec, CSc.**  
předseda oborové rady

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Cílem Bakalářské práce je seznámení se a zjištění možností řízení dozvukových vlastností zavedených algoritmů pro syntetickou modelaci dozvuku. Algoritmy, které navrhli Jon Dattorro, William Gardner a Jean-Marc Jot, jsou implementovány v prostředí Matlab. Jsou zjištěny jejich charakteristické vlastnosti a provedeno porovnání. Následná implementace Jotova algoritmu pomocí technologie VST v prostředí Matlab s využitím rozšíření Audio System Toolbox, umožňuje studovat vlastnosti a chování algoritmu, který zpracovává zvuk v reálném čase. Výsledný zásuvný modul lze poté testovat v DAW, které slouží jako hostitelská aplikace.

## KLÍČOVÁ SLOVA

Audio System Toolbox, Dattorrov reverberátor, fázovací článek, Gardnerův reverberátor, impulzní odezva, Jotův reverberátor, objektivní akustické parametry prostoru, umělý dozvuk, VST zásuvný modul, unitární zpětnovazební zpožďovací síť

## ABSTRACT

The aim of the Bachelor thesis is to get acquainted with and find out possibilities of control of the reverberation properties of established algorithms for creation of artificial room reverberation. The algorithms proposed by Jon Dattorro, William Gardner and Jean-Marc Jot are implemented in the Matlab environment. Their characteristics and comparison are determined. Subsequent implementation of Jot's algorithm into the VST plug-in module in Matlab using the Audio System Toolbox extension enables to study the properties and behavior of the algorithm that handles real-time sound. The resulting plug-in can then be tested in DAW as a host application.

## KEYWORDS

Audio System Toolbox, all-pass filter, Dattorro reverberator, Gardner reverberator, impulse response, Jot reverberator, objective acoustic parameters of an enclosed space, synthetic reverberation, VST plugin, unitary feedback delay network

BEŇO, Tomáš *Porovnání algoritmů pro syntetické modelování dozvukových vlastností místnosti*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 70 s. Vedoucí práce byl Ing. Petr Frenštátský

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Porovnání algoritmů pro syntetické modelování dozvukových vlastností místnosti“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Petru Frenštátskému za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)

# OBSAH

Úvod	10
<b>1 Teoretické poznatky</b>	<b>12</b>
1.1 Dozvuk	12
1.2 Přístup k návrhu reverberátorů	12
1.2.1 Fyzikální přístup	12
1.2.2 Perceptuální přístup	13
1.3 Dattorův reverberátor	14
1.3.1 Struktura topologie	14
1.4 Gardnerův reverberátor	18
1.4.1 Struktura topologie	18
1.5 Jotův reverberátor	22
1.5.1 Struktura topologie	22
1.6 Objektivní kritéria hodnocení poslechového prostoru	25
1.6.1 Doba dozvuku $RT$	25
1.6.2 Počáteční doba dozvuku $EDT$	25
1.6.3 Počáteční prodleva $ITDG$	26
1.6.4 Míra hlubokých tónů $BR$	26
1.6.5 Míra vysokých tónů $HFR$	26
1.6.6 Jasnost $C_{50}$	26
1.6.7 Míra přímého zvuku $C_7$	27
1.6.8 Míra jasnosti $C_{80}$	27
1.6.9 Zřetelnost $D_{50}$	27
1.6.10 Těžištvý čas $Ts$	28
1.6.11 Míra doznívání $H$	28
1.6.12 Interaurální koeficient křížové korelace $IACC$	28
1.6.13 Echo Criterion $EK$	28
<b>2 Implementace a měření algoritmů v Matlabu</b>	<b>29</b>
2.1 Implementace algoritmu Dattorova reverberátoru	31
2.2 Implementace algoritmu Gardnerova reverberátoru	35
2.3 Implementace algoritmu Jotova reverberátoru	38
<b>3 Implementace Jotova reverberátoru pomocí technologie VST</b>	<b>41</b>
3.1 Vývojové prostředí	41
3.1.1 Nástroj Audio Test Bench	41
3.1.2 Doporučený pracovní postup	42

3.1.3	Třída <code>AudioPlugin</code> . . . . .	42
3.2	Návrh algoritmu . . . . .	44
3.2.1	Kruhová vyrovnávací paměť . . . . .	45
3.3	Implementace . . . . .	46
3.3.1	Sekce <i>Vlastnosti</i> . . . . .	46
3.3.2	Grafické rozhraní . . . . .	46
3.3.3	Funkce <i>konstruktor</i> . . . . .	47
3.3.4	Funkce <i>reset</i> . . . . .	47
3.3.5	Funkce <i>set</i> . . . . .	47
3.3.6	Funkce <i>process</i> . . . . .	48
3.4	Testování a měření . . . . .	49
3.4.1	Testování v Audio Test Bench . . . . .	50
3.4.2	Testování v hostitelské aplikaci . . . . .	51
<b>4</b>	<b>Výsledky práce</b>	<b>54</b>
4.1	Vyhodnocení implementace algoritmů v prostředí Matlab . . . . .	54
4.1.1	Dattorův reverberátor . . . . .	54
4.1.2	Gardnerův reverberátor . . . . .	54
4.1.3	Jotův reverberátor . . . . .	55
4.1.4	Výsledek porovnání . . . . .	56
4.2	Vyhodnocení implementace Jotova algoritmu pomocí technologie VST	57
4.2.1	Shrnutí možností rozšíření Audio System Toolbox . . . . .	57
4.2.2	Výsledek implementace pomocí technologie VST . . . . .	57
4.2.3	Možnosti dalšího rozšíření . . . . .	58
	<b>Závěr</b>	<b>60</b>
	<b>Literatura</b>	<b>61</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>64</b>
	<b>Seznam příloh</b>	<b>66</b>
	<b>A Vstupní parametry reverberátorů</b>	<b>67</b>
	<b>B Obsah přiloženého DVD</b>	<b>70</b>



# SEZNAM OBRÁZKŮ

1.1	Znázornění průběhu dozvuku v čase. . . . .	12
1.2	Zjednodušená struktura tanku. . . . .	15
1.3	Extrakce signálu z vnitřní struktury fázovacího článku. . . . .	16
1.4	Schéma Dattorrova plate reverberátoru. . . . .	17
1.5	Zjednodušené schéma recirkulační smyčky. . . . .	19
1.6	Schéma jednoduchého vnořeného fázovacího článku. . . . .	20
1.7	Schéma Gardnerova reverberátoru v konfiguraci Velká místnost. . . .	21
1.8	Schéma Jotova UFDN reverberátoru. . . . .	24
2.1	Schéma filtru typu dolní propust. . . . .	30
2.2	Schéma fázovacího článku. . . . .	30
2.3	Impulzní odezva Dattorrova reverberátoru. . . . .	34
2.4	Reliéf poklesu energie Dattorrova reverberátoru. . . . .	34
2.5	Impulzní odezva Gardnerova reverberátoru. . . . .	37
2.6	Reliéf poklesu energie Gardnerova reverberátoru. . . . .	37
2.7	Impulzní odezva Jotova reverberátoru. . . . .	40
2.8	Reliéf poklesu energie Jotova reverberátoru. . . . .	40
3.1	Princip kruhové vyrovnávací paměti. . . . .	45
3.2	Grafické prostředí nástroje <i>Audio Test Bench</i> . . . . .	50
3.3	Nativní rozhraní zásuvného modulu v hostitelské aplikaci <i>FL Studio</i> . .	51
3.4	Impulzní odezva Jotova reverberátoru implementovaného ve VST. . .	53
3.5	Reliéf poklesu energie Jotova reverberátor implementovaného ve VST. .	53

# SEZNAM TABULEK

1.1	Výchozí typy konfigurací Gardnerova reverberátoru . . . . .	18
2.1	Seznam hodnot sloužících k vytvoření stereofonního výstupu. . . . .	31
2.2	Objektivní akustické parametry Dattorrova reverberátoru. . . . .	33
2.3	Objektivní akustické parametry Gardnerova reverberátoru. . . . .	36
2.4	Objektivní akustické parametry Jotova reverberátoru. . . . .	39
3.1	Objektivní akustické parametry Jotova reverberátoru implementova- ného ve VST. . . . .	52
4.1	Porovnání doby provádění výpočtu dozvuku pro jeden vzorek. . . . .	56
A.1	Vstupní parametry Dattorrova reverberátoru. . . . .	67
A.2	Vstupní parametry Gardnerova reverberátoru. . . . .	68
A.3	Vstupní parametry Jotova reverberátoru. . . . .	69

# ÚVOD

Jedním z nejzásadnějších fenoménů v akustice je dozvuk. Dozvuk je výsledkem mnoha odrazů v uzavřeném prostoru, které vznikají šířením zvuku od zdroje k posluchači. Signál vyslaný zvukovým zdrojem do prostoru jednak zamíří nejkratší cestou k posluchači jako přímá vlna, ale také ostatními směry, dokud nenarazí na překážku a odrazí se zpět. Takto odražená zvuková vlna dorazí zpět k posluchači vlivem velikosti a geometrie místnosti, povrchu a členitosti s určitým časovým zpožděním, menší intenzitou a změněnou fází. Odraz vln probíhá v mnohem větším měřítku a posluchač jejich součet vnímá jako vjem prostoru, lépe lokalizuje zdroje zvuku a přibližnou velikost a tvar místnosti.

S dozvukem se setkáváme denně a přijde nám jako přirozená informace o našem okolí. Dá se také využít s uměleckým záměrem v hudbě pro prostorové umístění jednotlivých nástrojů případně celé skladby. Kromě toho, většina nahrávek pochází ze studiového prostředí, kde je dozvuk záměrně snížený, a proto zní poněkud nepřirozeně. Z toho důvodu je dozvuk je velmi žádoucím v hudebním a také filmovém průmyslu jako zvukový efekt. Požadavek na stále větší potřebu umělého dozvučování a vývoj elektrotechniky v 50. a 60. letech minulého století vedl k experimentování s elektronickými obvody pro simulaci dozvuku. Takto vytvořený dozvuk se nazývá umělý nebo syntetický dozvuk a zařízení nebo aplikace, která umí generovat dozvuk, se nazývá reverberátor (slangově *reverb*).

Prvním průkopníkem byl v roce 1961 Robert Schröder s reverberátorem na bázi paralelních hřebenových filtrů následovaných sérií fázovacích článků [1]. Jeho návrh v sedmdesátých letech vylepšil James Moorer. Přidal zpožďovací linku, která se nazývá *Tapped delay line* (*TDL*), a umístil ji před hřebenové filtry, aby simuloval prvotní odrazy. Také přidal filtr typu dolní propust do zpětnovazební smyčky hřebenových filtrů, aby simuloval absorpční charakteristiku vzduchu, která je charakteristická útlumem energie na vyšších kmitočtech [2]. V devadesátých letech vznikly různé algoritmy, které se snaží odstranit nedostatky reverberátorů založených na hřebenových filtrech. Mají propracovanější možnosti úpravy a řízení parametrů, které značně upravují výslednou odezvu.

Cílem této práce je seznámení se právě s těmito algoritmy. Konkrétně se jedná o ty, které navrhl Jon Dattorro, William Gardner a Jean-Marc Jot. Následná implementace algoritmů realizujících umělý dozvuk v prostředí *Matlab* umožňuje zobrazit, změřit a zaznamenat jejich charakteristické vlastnosti. Dále je Jotův algoritmus implementován pomocí technologie VST přímo v prostředí *Matlab* s využitím rozšíření, které se nazývá *Audio System Toolbox*. Práce se nezabývá hledáním parametrů pro dosažení ideálního výsledku, protože neexistuje obecná metoda k získání dostatečně dobře znějícího výstupu, který by se blížil kvalitou profesionálním aplikacím.

Precizní nastavení je časově velmi náročné a úspěch není dopředu zaručen.

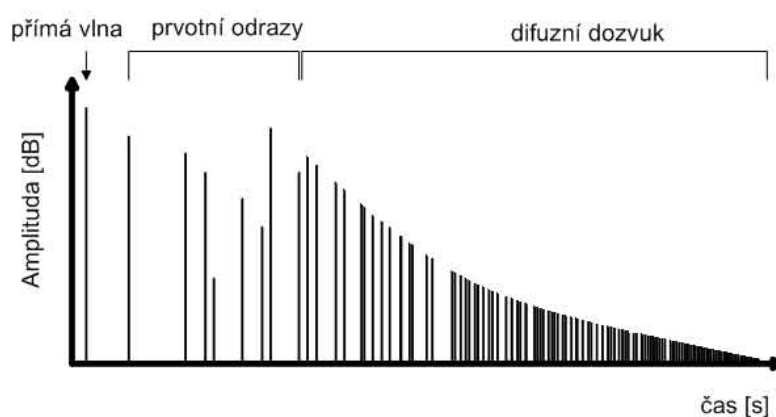
První část dokumentu se věnuje teoretickým poznatkům k umělému dozvuku, obecným informacím k jednotlivým reverberátorům a popisu kritérií pro objektivní hodnocení akustických parametrů místnosti. V druhé části je popsána implementace jednotlivých algoritmů v *Matlabu*. Třetí část se věnuje implementaci Jotova algoritmu pomocí technologie VST, popisu a zjišťování možností *Audio System Toolboxu* a testování VST zásuvného modulu v DAW a za pomoci nástroje *Audio Test Bench*. V Další kapitole je provedeno porovnání, zhodnocení a jsou sumarizovány zjištěné poznatky k možnostem řízení dozvukových vlastností všech implementovaných algoritmů.

# 1 TEORETICKÉ POZNATKY

## 1.1 Dozvuk

Dozvuk lze v časové rovině rozdělit na prvotní odrazy a difúzní dozvuk. Prvotní odrazy mají převážně diskrétní podobu, protože počet odrazů je dobře, kdy je zaznamenaná posluchač, nízký. Tyto odrazy mají vliv na přímý zvuk a jeho hlasitost, barvu a prostorový vjem [3].

Postupem času se počet odrazů zvyšuje a stejně tak klesá amplituda a energie zvuku na vysokých kmitočtech, jak je možno vidět na obr. 1.1. Hustota těchto odrazů putujících všemi směry odpovídá definici difúzního zvukové pole.



Obr. 1.1: Znázornění průběhu dozvuku v čase.

## 1.2 Přístup k návrhu reverberátorů

Při návrhu reverberátoru existuje více přístupů k získání syntetického dozvuku a na základě požadavků a priorit je důležité zvolit vhodný přístup. Obecně existují dva. Fyzikální a perceptuální neboli vjemový.

### 1.2.1 Fyzikální přístup

Fyzikální přístup se snaží dosáhnout stejného šíření zvukových vln, jako v dané místnosti. Možnost, jak toho dosáhnout, je konvoluce vstupního signálu se změřenou nebo syntetickou impulzní odezvou. Jedná se v podstatě o filtr s konečnou impulzní odezvou. Konvoluční reverberátory jsou vhodné pro věrné napodobení existujících koncertních sálů, které mají požadované akustické vlastnosti. Jejich použití se stalo

běžné až posledních letech, kdy výpočetní výkon počítačů zvládne bez problémů konvoluci dlouhých impulzních odezev v reálném čase s přijatelnou latencí.

Na druhou stranu fyzikální přístup spoléhá ve své definici na těžkopádnou reprezentaci v podobě impulzní odezvy, pro kterou je změna jejich parametrů složitá a neefektivní [4]. K simulaci obvyklých prostorů potřebuje konvoluce signálu v reálném čase velký výpočetní výkon, protože impulzní odezva obsahuje velké množství informací. Ačkoliv je přesná simulace v některých případech nezbytná, například pro potlačení ozvěny, pro získání věrohodného vjemu dozvukových vlastností místností není fyzikální přístup nutný.

### 1.2.2 Perceptuální přístup

Perceptuální přístup vychází ze základního požadavku na snížení výpočetní náročnosti algoritmu. Nesimuluje se tak přímo impulzní odezva akustického prostoru, ale pouze specifické vlastnosti akustického prostoru. Tyto specifické vlastnosti jsou definovány jako lidské vjemy a parametrizovány [5].

Mezi přednosti reverberátorů založených na perceptuálním přístupu patří velká variabilita a množství parametrů, které lze ovládat, a také možnost použití na širokém spektru zvukového materiálu. Nezanedbatelnou předností je také často zlomek výpočetního výkonu potřebný k dosažení podobných vlastností dozvuku, jako při konvoluci. Cenou za tento způsob je oproti konvolučním reverberátorům snížená věrohodnost prostoru a zvýšená pravděpodobnost výskytu nežádoucích sonických artefaktů.

## 1.3 Dattorrův reverberátor

Dattorrův reverberátor je založený na principu kaskádně zapojených fázovacích článků. Tato topologie je nejmenší rekurzivní síť z pohledu výpočetní náročnosti a složitosti algoritmu, zároveň ale splňuje subjektivní požadavky na dobře znějící dozvuk. Teoreticky může existovat nekonečně mnoho variací. Ještě před Dattorrem se tomuto tématu věnovali Griesinger a Blesser, ale jejich práce na výzkumu nebyla dostatečně zdokumentovaná [6]. Mezi přednosti této topologie patří ovládací prvky, pomocí kterých lze řídit konkrétní aspekty dozvuku, jako *vstupní difúze*, *difúze poklesu*, *pokles* a *útlum*, ačkoliv některé prvky nejsou přímo vztažené k fyzikálním vlastnostem dozvuku. Navíc dozvuk je vhodný pro širokou škálu zdrojových signálů [7].

Tento reverberátor je z kategorie tzv. *plate* reverberátorů. Tato kategorie je charakteristická téměř okamžitým nárůstem hustoty odrazů s hladkou exponenciálně klesající amplitudovou obálkou průběhu dozvuku. Také se vyznačuje kratší maximální dobou dozvuku ve srovnání s jinými typy algoritmů pro syntetické modelování dozvukových vlastností místnosti. Je třeba vzít do úvahy, že pro rychlý nástup odrazů je spíše kontraproduktivní použít modul, který vytváří prvotní odrazy. Proto i Dattorro při návrhu počítal jen s algoritmem pro tvorbu difúzního dozvuku [6].

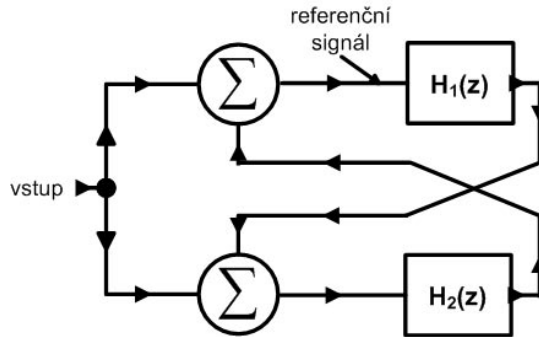
### 1.3.1 Struktura topologie

Signál je na vstupu nejprve zpožděn prvkem počáteční zpoždění, který zpozdí vstupní signál o nastavený čas v milisekundách parametrem *počáteční zpoždění*. Počáteční zpoždění představuje dobu, kdy k nám dorazí první odražená vlna. To znamená, že vyšší hodnoty přispívají k dojmu velkého prostoru. Takto zpožděný signál je filtrován v závislosti na nastavení parametrem *mezní kmitočet* filtrem typu dolní propust, který simuluje absorpční ztráty, které vznikají poklesem energie na vyšších kmitočtech.

Následuje *dekorelační úsek* složený ze čtyř fázovacích článků zapojených do série, jejichž funkce je rychle dekorelovat vstupní signál před vstupem do struktury *tank*. Při pohledu na schéma na obr. 1.4 lze tyto úseky vidět označené jako *dekorelační úsek* a *tank*. Tyto úseky představují stěžejní části celé topologie. Recirkulace signálu v *tanku* se může stát slyšitelnou, pokud není signál na vstupu dostatečně dekorelován. Tato funkce nabývá na důležitosti při aplikaci dozvuku na zvuky perkusivního charakteru. Dekorelaci na vstupu lze také vnímat jako vytvoření náhodné fáze signálu, která vyhladí špičky a další nepravidelné útvary [1]. Celý úsek je řízený pomocí parametrů *vstupní difúze 1* a *vstupní difúze 2*, které ve skutečnosti nastavují zesílení

fázovacích článků 1 a 2, resp. 3 a 4. Dekorelovaný výstup je rozdělený do dvou symetrických větví a vstupuje do smyčky, kterou Dattorro nazval *tank*. Symetrických ve smyslu stejného počtu a pořadí funkčních bloků. Koeficienty fázovacích článků a zpoždění jsou v každé větvi odlišné. Recirkulační struktura *tank* působí především na vlastnosti difúzní oblasti oblasti dozvuku. Účelem zde implementovaných fázovacích článků je eliminovat jakýkoliv opakující se zvukový vzorek. Úspěšnost této operace je závislá na charakteru vstupního signálu. Nastavení těchto fázovacích článků je důležité pro získání exponenciálně klesající křivky obálky impulzní odezvy; vše musí být nastaveno na základě subjektivních rozhodnutí [6]. Kromě nastavení ovládacích prvků pokles difúze 1 a 2, lze nastavit útlum parametrem *útlum*, který ovládá mezní kmitočet LP filtrů v každé větvi. Na koncích jednotlivých větví se nachází parametr *decay*, realizovaný jednoduchým násobením výstupu konstantou. Parametr *decay* má zásadní vliv na tvar impulsové odezvy. Na obr. 1.2 je zjednodušená struktura *tanku* s vyznačením vstupu a referenčního signálu, jejichž výpočet je možný na základě znalosti přenosových funkcí  $H_1(z)$ ,  $H_2(z)$  a  $T(z)$ , viz rovnice

$$T(z) = \frac{\text{vstup}}{\text{referenční signál}} = \frac{1 + H_2(z)}{1 - H_1(z)H_2(z)}. \quad (1.1)$$



Obr. 1.2: Zjednodušená struktura tanku.

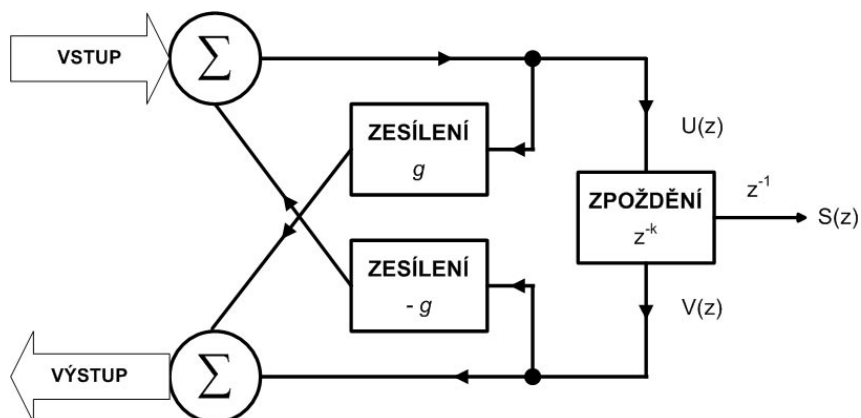
Výstupy jsou tvořené z několika výstupních bodů uvnitř tanku. Z každého bodu se odebírá signál zvlášť pro levý a pro pravý kanál. Navíc s předepsanou polaritou a velikostí zpoždění uvedených v tab. 2.1. Ve dvou bodech probíhá extrakce výstupního signálu přímo z vnitřní struktury fázovacího článku. Schématické znázornění extrakce signálu je na obr. 1.3. Pro extrahovaný signál platí:

$$V(z) = z^{-k}U(z), U(z) = X(z) - gV(z), S(z) = z^l - 1)U(z), \quad (1.2)$$

kde  $S(z)$  představuje požadovaný výstup. Jeho přenosová funkce vypadá následovně:

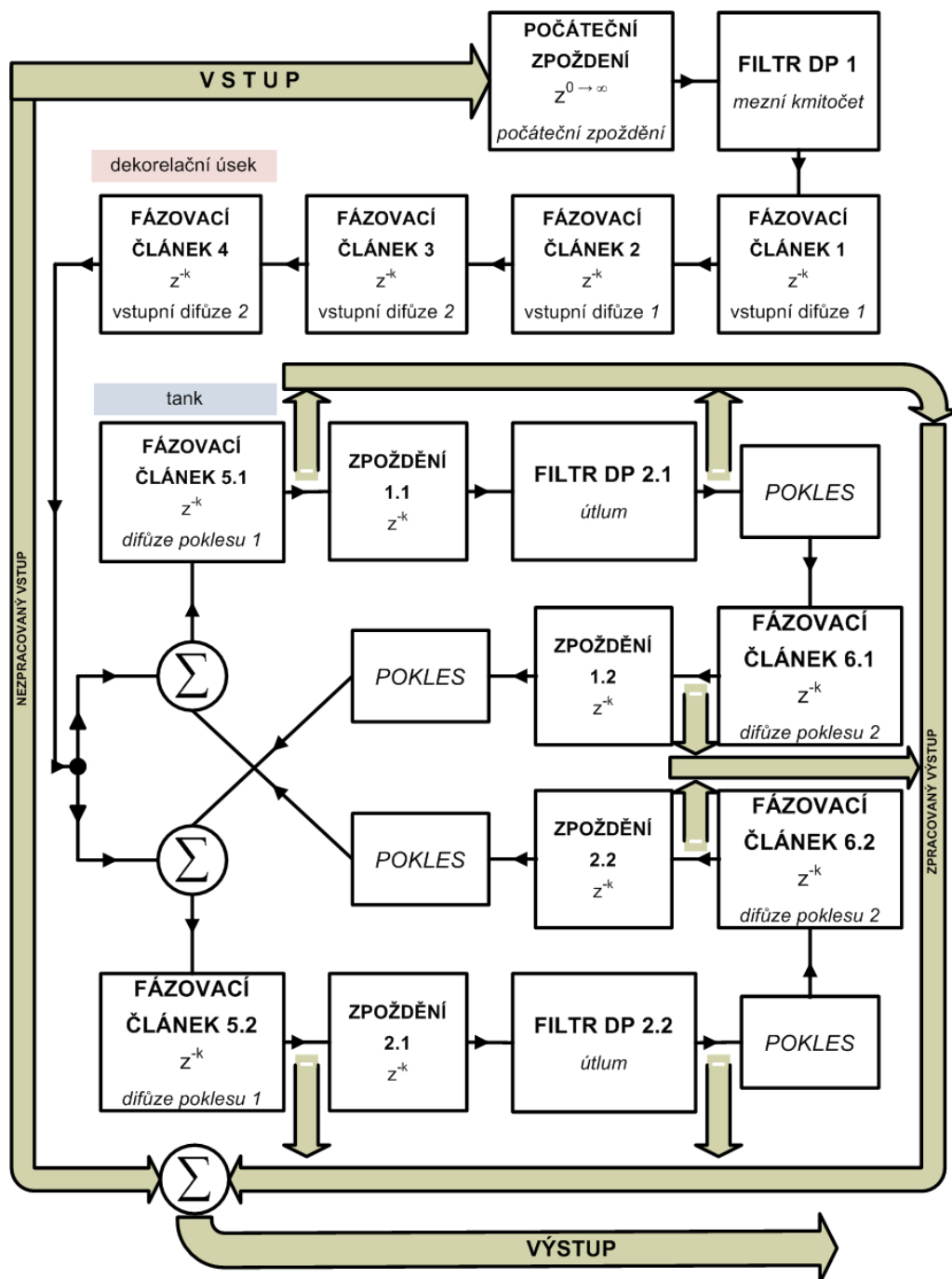


$$S(z) = \frac{z^{-1}X(z)}{1 + gz^{-k}}. \quad (1.3)$$



Obr. 1.3: Extrakce signálu z vnitřní struktury fázovacího článku.

Získané zpracované stereofonní signály jsou na výstupu sečteny s nezpracovaným signálem v požadovaném poměru.



Obr. 1.4: Schéma Dattorrova plate reverberátoru.

## 1.4 Gardnerův reverberátor

Princip Gardnerova reverberátoru spočívá v implementaci vnořených fázovacích článků. Tuto myšlenku poprvé vyslovil v osmdesátých letech Vercoe [9]. Jedná se o vložení fázovacího článku do zpožďovací struktury jiného fázovacího článku, viz 1.6. Takto lze dosáhnout oproti jednoduchému fázovacímu článku exponenciálního nárůstu odrazů, stejně jako v reálných místnostech.

Gardner navrhl pro svou topologii tři výchozí konfigurace. *Small room*, *Medium room* a *Large room*. Jedná se o menší změny uvnitř struktury, například jinou posloupností fázovacích článků a odlišným nastavením koeficientů. To proto, že pokud se například při konfiguraci *Small room* nastaví dlouhá doba dozvuku, výsledek je neuspokojivý, a stejně tak to platí pro krátký dozvuk u konfigurace *Large room* [10]. Pro tuto práci byla zvolena konfigurace *Large room*, jejíž struktura je na obr. 1.7. Vzhledem k tomu, že Gardnerova prvotní implementace v rámci jeho Diplomové práce byla zaměřena na prostorové ozvučení místnosti, tzv. *surround* se speciální implementací prvotních odrazů, je výstup tohoto reverberátoru pouze monofonní.

Tab. 1.1: Výchozí typy konfigurací Gardnerova reverberátoru

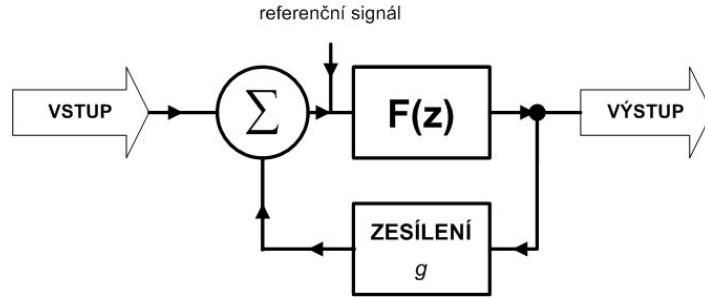
konfigurace	rozsah doby dozvuku [s]
Malá místnost (Small Room)	0,38-0,57
Střední místnost (Medium Room)	0,58-1,29
Velká místnost (Large Room)	1,3- $\infty$

### 1.4.1 Struktura topologie

Signál na vstupu je zpožděn o hodnotu parametru *počáteční zpoždění* a pokračuje do bloku nazvaného prvotní odraz. Ty jsou realizované pomocí jednoduché zpožďovací linky bez rekurzivní struktury, tzn. že jedná se o šest filtrů typu zpoždění s konečnou impulzní charakteristikou FIR. V angličtině se tato zpožďovací struktura nazývá *tapped delay line*, zkráceně TDL. Na konci tohoto bloku je filtr dolní propust, který simuluje absorpci zvuku prostředím. Výstup je vyveden na vstup difúzní sekce reverberátoru a také na výstup, kde se sečte s nezpracovaným signálem a výstupem difúzní sekce.

Jádrem celého algoritmu je recirkulační smyčka obsahující samostatné i vnořené fázovací články a několik modulů zpoždění, ze kterých je na různých místech odebírán výstupní signál. Na konci smyčky se nachází zpětnovazební zesílení  $g$ , které zajišťuje, že bude energie signálu po každém průchodu smyčkou klesat. Na obr. 1.5

je schéma recirkulační smyčky, kde jsou veškeré zpožďovací prvky zahrnuty do přenosové funkce  $F(z)$ .



Obr. 1.5: Zjednodušené schéma recirkulační smyčky.

Přenosová funkce celého reverberátoru je

$$G(z) = \frac{F_d(z)}{F_d(z) - gF_n(z)}, \text{ kde } F(z) = \frac{F_n(z)}{F_d(z)}. \quad (1.4)$$

Koeficient  $g$  je zpětnovazební zesílení,  $F_n(z)$  čitatel a  $F_d(z)$  jmenovatel přenosové funkce  $F(z)$  [7]. K získání obecné přenosové funkce  $G(z)$  musíme nejprve spočítat čitatele a jmenovatele přenosové funkce. Dále můžeme vyjádřit přenosovou funkci vnořeného fázovacího članku jako

$$H(z) = \frac{-kF_d(z) + F_n(z)}{F_d(z) - kF_n(z)}, \quad (1.5)$$

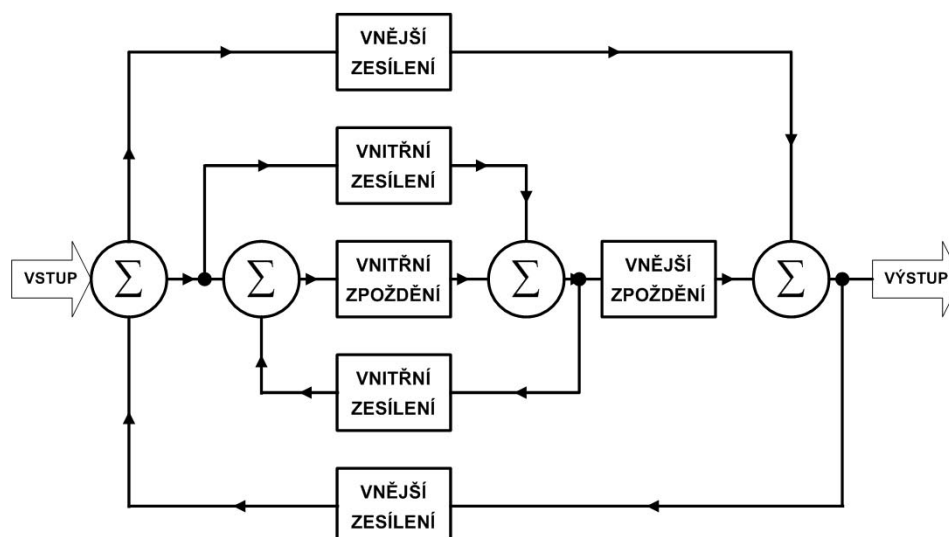
kde  $k$  představuje zesílení vnějšího fázovacího članku.

Obecné schéma jednoduchého vnořeného fázovacího članku je na obr. 1.6. Nutno poznamenat, že zesílení ve zpětnovazebních větvích je záporné.

Výstupy, jejichž celkový počet je 3, jsou váhované koeficientem zesílení a poté sečteny dohromady. Vzhledem k tomu, že jednotlivé výstupy jsou odebírány v různých bodech, každý z nich má jinou křivku dozvuku. Vhodným váhováním koeficientů zesílení výstupů je možné upravit charakter výstupu celého algoritmu.

Odrazy generované vnitřním fázovacím článkem recirkulují přes celou zpětnovazební smyčku zpět na své vstupy, a proto v závislosti na čase roste počet odrazů. U jednoduchých fázovacích článků je jejich počet v čase spíše konstantní [10]. Odezva sériově umístěných i vnořených fázovacích článků je stejná a není důvod obávat se vzniku nestability.

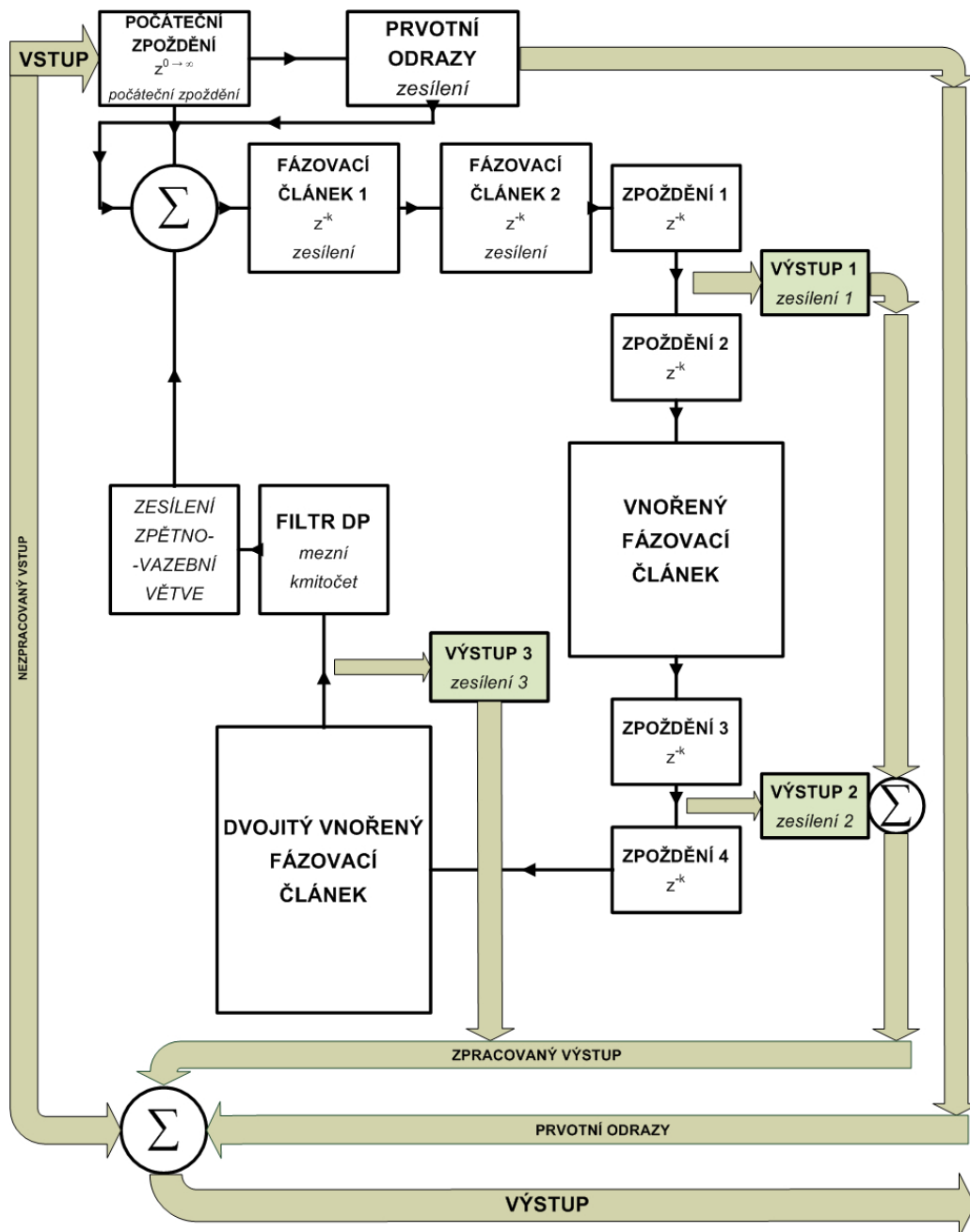
Teoreticky lze použít i vnořený hřebenový filtr, ale odezva v kmitočtové oblasti by vykazovala známky rezonance a mohla by vzniknout nestabilita systému. Byla by



Obr. 1.6: Schéma jednoduchého vnořeného fázovacího článku.

chyba předpokládat, že se nemůže objevit tonální zabarvení výstupu, protože kmitočtová charakteristika má neovlivněný průběh. To proto, že náš sluchový aparát pracuje způsobem, díky kterému může zaznamenat občasné zabarvení, které připomíná kovové zvonění či bzučení, přestože v delším časovém měřítku jsou všechny kmitočty vyvážené.

Podle Gardnera existuje nepřehledné množství možných variací reverberátorů postavených na základě této topologie. Přesto byla tato topologie navržena experimentální metodou, protože trvalo získat vhodné seskupení a nastavení parametrů. Pokud by byla realizována automatická metoda, která by vypočítala impulzní odezvu reverberátoru na základě požadovaných atributů, potom by mohla být navržena nelineárními technikami vyhledávání, jako *gradient descent*, *simulated annealing* nebo za použití *genetických algoritmů*. Gardner sám experimentoval s genetickými algoritmy pro návrh reverberátoru. Ačkoliv změřené hodnoty vypadaly dobře, charakter dozvuku byl nepříjemný. Obecně je problém navrhnout algoritmus, který vyhodnocuje dozvuk stejným způsobem, jako lidský mozek [10].



Obr. 1.7: Schéma Gardnerova reverberátoru v konfiguraci Velká místnost.

## 1.5 Jotův reverberátor

Jotův reverberátor je v principu založený na zpětnovazební zpožďovací síti s útlumovými členy. Zpětnovazební zpožďovací síť (FDN - feedback delay network) je LTI systém, který zachovává celkovou energii všech vstupních signálů. Impulzní odezva tohoto systému připomíná stacionární bílý šum [11]. Zpětnovazební zpožďovací síť je zobecněním Schröderova paralelního zapojení hřebenových filtrů, ale dosahuje větší hustoty odrazů.

Produkt dvou unitárních matic je unitární a stejně tak platí, že kaskáda dvou unitárních systémů je stále unitární systém [12]. Pokud se vezme do úvahy, že zpožďovací linky jsou unitární systém a matice ve zpětné vazbě je zvolena tak, že je také unitární, pak lze takový systém nazývat unitární zpětnovazební zpožďovací síť. Podmínku unitární matice splňuje matice, kterou použili Stautner a Puckette v [13]. Pokud jsou koeficienty unitární matice reálná čísla, transponovaná matice se rovná inverzní matici původní matice. Tato definice odpovídá definici ortogonální matice. Matice  $\mathbf{A}$  je ortogonální, pokud platí:

$$\mathbf{A}\mathbf{A}^T = \mathbf{I}, \quad (1.6)$$

kde  $\mathbf{I}$  je jednotková matice a  $\mathbf{A}^T$  transponovaná matice  $\mathbf{A}$ . Tento vztah platí, pouze pokud jsou koeficienty matice reálné [12].

Zpětnovazební matice umožňuje výstupům každé zpožďovací linky ovlivnit vstupy všech vstupů zpožďovacích linek. Koeficient zesílení této matice váhuje průchozí signál. Obecně platí, že pokud jsou prvky na hlavní diagonále matice nulové, výstupy zpožďovacích linek nebudou ovlivňovat své vlastní vstupy. Koeficienty nesmí být větší než 1 z důvodu stability. Mezi obecně známé typy matic, které splňují tyto předpoklady patří Hadamardova nebo Householderova matice.

### 1.5.1 Struktura topologie

Jot původní zpětnovazební zpožďovací síť vylepšil přidáním absorpčního zesílení  $k_i$  pro každou zpožďovací linku, které simuluje absorpci zvuku následkem odrazů od stěn a odporu částic vzduchu [14]. Přidal také zesílení na vstupech a výstupech jednotlivých linek a filtr tonální korekce, který je realizován jako filtr typu dolní propust. Pro výpočet absorpčního zesílení kanálu platí vztah

$$g_i = 10^{-\frac{3k_i T_s}{RT}}, \quad (1.7)$$

kde  $T_s$  je vzorkovací perioda a  $RT$  odpovídá hodnotě parametru *doba dozvuku*.

Přenosová funkce Jotova reverberátoru je

$$H(z) = \frac{\mathbf{Y}(z)}{\mathbf{X}(z)} = \frac{\mathbf{k}z^{-d}}{\mathbf{I} - \mathbf{A}\mathbf{k}z^{-d}}, \quad (1.8)$$

kde  $z^{-d}$  a  $\mathbf{k}$  jsou diagonální matice pro zpoždění a zesílení

$$z^{-d} = \begin{pmatrix} z^{-d_1} & & & \\ & z^{-d_2} & & \\ & & z^{-d_3} & \\ & & & z^{-d_4} \end{pmatrix}, \quad (1.9)$$

$$\mathbf{k} = \begin{pmatrix} k_1 & & & \\ & k_2 & & \\ & & k_3 & \\ & & & k_4 \end{pmatrix}. \quad (1.10)$$

$\mathbf{X}(z)$  a  $\mathbf{Y}(z)$  jsou sloupkové matice pro vstupní a výstupní signál

$$\mathbf{X}(z) = \begin{pmatrix} X_1(z) \\ X_2(z) \\ X_3(z) \\ X_4(z) \end{pmatrix}, \quad (1.11)$$

$$\mathbf{Y}(z) = \begin{pmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \\ Y_4(z) \end{pmatrix}. \quad (1.12)$$

$\mathbf{A}$  je zpětnovazební matice

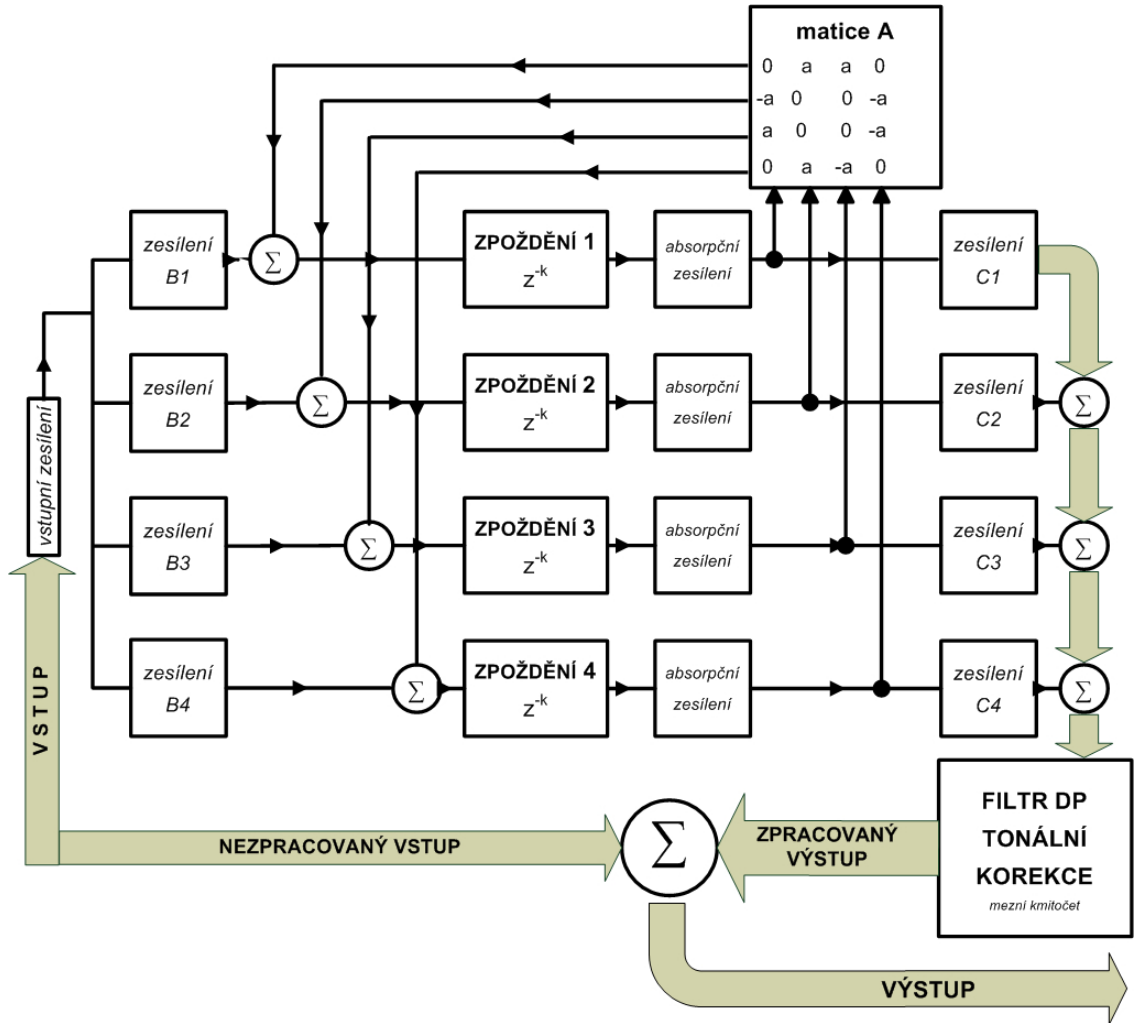
$$\mathbf{A} = \frac{\sqrt{2}}{2} \begin{pmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix} \quad (1.13)$$



a  $\mathbf{I}$  jednotková matice

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.14)$$

Existují práce, které se věnují tématu rozšíření Jotova algoritmu, například [15], které navrhuji způsoby, na základě kterých lze vytvořit koeficienty vedoucí k dobře znějícímu výsledku [8]. Další práce se zabývají automatizací vytvoření impulsní odezvy pomocí genetických algoritmů. Jot později rozšířil svůj algoritmus tak, že nahradil absorpční zesílení frekvenčně závislými absorpčními filtry pro každou zpožďovací linku. Na následujícím obrázku se nachází schéma topologie Jotova reverberátoru.



Obr. 1.8: Schéma Jotova UFDN reverberátoru.

## 1.6 Objektivní kritéria hodnocení poslechového prostoru

Objektivní kritéria hodnocení poslechového prostoru představují sadu parametrů popisující akustické vlastnosti prostoru. V principu lze kritéria poslechového prostoru rozdělit na časová, frekvenční, energetická a ostatní kritéria. Mezi časová patří doba dozvuku, či počáteční doba dozvuku. K frekvenčním náleží míra hlubokých a vysokých kmitočtů. Mezi energetická patří například jasnost a míra přímého zvuku. Časová a frekvenční kritéria se stanovují z časového úseku křivky poklesu zvukové energie. Energetická kritéria vychází z energetických poměrů signálů pro definované časové úseky průběhů získané z impulzní odezvy prostoru. Pomocí těchto parametrů lze zjistit na základě impulzní odezvy také vlastnosti algoritmů simulující dozvuk.

### 1.6.1 Doba dozvuku $RT$

Abychom mohli charakterizovat dozvuk kvantitativně, definujeme dobu dozvuku jako dobu, za kterou klesne hustota zvukové energie nebo intenzita zvuku po vypnutí zvukového zdroje na  $10^{-6}$  původní hodnoty, tj. pokles o 60 dB. Pro standardní dobu dozvuku používáme značení  $T_{60}$  ( $RT$ ). Je určena přímkou se směrnici  $b$ , proloženou poklesem dozvuku  $D(t)$  v rozmezí mezi hladinami  $D = -5$  dB až  $D = -35$  dB

$$T_{60} = -\frac{60}{b}. \quad (1.15)$$

Dozvukový pokles  $D(t)$  je definován jako

$$D(t) = 10 \log\left(1 - \frac{E_t}{E_\infty}\right). \quad (1.16)$$

### 1.6.2 Počáteční doba dozvuku $EDT$

Jak počáteční doba dozvuku  $EDT$ , tak standardní doba dozvuku  $RT$  se získávají měřením ze sklonu křivek oktávového pásma integrované impulsové odezvy. Sklon křivky poklesu by se měl určit ze sklonu regresní přímky proložené příslušnou částí křivky poklesu. Počáteční doba dozvuku  $EDT$  se získává z počátečních 10 dB poklesu a doba dozvuku  $RT$  se získává z části poklesu mezi úseky  $-5$  dB a  $-35$  dB pod maximální počáteční hladinou. Doby dozvuku se z těchto sklonů vypočítají jako doby potřebné k poklesu o 60 dB. Veličina  $EDT$  je subjektivně důležitější a více souvisí s vnímaným dozvukem prostoru, zatímco  $RT$  se týká fyzikálních vlastností poslechového prostoru.

### 1.6.3 Počáteční prodleva $ITDG$

Počáteční prodleva  $ITDG$  je časové zpoždění prvních odrazů od bočních stěn a stropu za přímým zvukem s odrazy nejblížešších ploch. Hodnota  $ITDG$  je pro celý sál určována z impulsové odezvy obvykle poblíž středu přízemí na jeho podélné ose. Zdroj je umístěn ve středu podélné osy pódia, obvykle 1,5 až 2 m od jeho přední hrany. Upřednostňovány jsou hodnoty menší než 20 ms.

### 1.6.4 Míra hlubokých tónů $BR$

Míra hlubokých tónů je míra podpory hlubokých tónů dozívání sálu. Určí se z poměrů dob dozvuku v oktavových pásmech 125 a 250 Hz k době dozvuku v oktavových pásmech 500 a 1000 Hz.

$$BR = \frac{RT_{125} + RT_{250}}{RT_{500} + RT_{1000}}, \quad (1.17)$$

kde  $RT_{125}$  je doba dozvuku v oktavovém pásmu 125 Hz,  $RT_{250}$  je doba dozvuku v oktavovém pásmu 250 Hz,  $RT_{500}$  je doba dozvuku v oktavovém pásmu 500 Hz,  $RT_{1000}$  je doba dozvuku v oktavovém pásmu 1000 Hz. Hodnoty  $BR$  mají být blízké 1, nikdy však nemají klesnout pod tuto hodnotu.

### 1.6.5 Míra vysokých tónů $HFR$

$$HFR = \frac{RT_{500} + RT_{1000}}{RT_{2000} + RT_{4000}}, \quad (1.18)$$

kde  $RT_{500}$  je doba dozvuku v oktavovém pásmu 500 Hz,  $RT_{1000}$  je doba dozvuku v oktavovém pásmu 1000 Hz,  $RT_{2000}$  je doba dozvuku v oktavovém pásmu 2000 Hz,  $RT_{4000}$  je doba dozvuku v oktavovém pásmu 4000 Hz. Hodnoty  $HFR$  mají být blízké 1, nikdy však nemají klesnout pod tuto hodnotu.

### 1.6.6 Jasnost $C_{50}$

Tato hodnota udává akustické kritérium srozumitelnosti pro mluvené slovo. Je definována poměrem energie přicházející do 50 ms k energii přicházející po 50 ms a její hodnota je v decibelech

$$C_{50} = 10 \log \frac{\int_0^{50ms} p^2(t) dt}{\int_{50ms}^{\infty} p^2(t) dt} = 10 \log \frac{E_{50}}{E_{\infty} - E_{50}}. \quad (1.19)$$

Každá hodnota nad 0 dB v standardní místnosti ukazuje na dobrou srozumitelnost mluveného slova zdroje zvuku. Hodnoty pod  $-5$  dB ukazují na více odraznou místnost než by byla vhodná pro přednes mluveného slova.

### 1.6.7 Míra přímého zvuku $C_7$

Je definována poměrem energií přicházejících do 7 ms k energii od 7 ms. Její hodnota je udávána v decibelech a používá se k predikci síly přímého zvuku zdroje (lokalizace přímého zvuku)

$$C_7 = 10 \log \frac{\int_0^{7ms} p^2(t) dt}{\int_{7ms}^{\infty} p^2(t) dt} = 10 \log \frac{E_7}{E_{\infty} - E_7}. \quad (1.20)$$

Hodnoty nad  $-15$  dB připouští dobrou lokalizaci zvuku zdroje. Čím více se hodnoty blíží 0 dB tím lepší je hodnota lokalizace zvuku zdroje.

### 1.6.8 Míra jasnosti $C_{80}$

Zakotvená v normě ČSN 73 0525 udává akustické kritérium pro hudbu (definována pro střed oktavového pásma 1000 Hz) a je vypočítána desetinásobkem dekadického logaritmu poměru zvukové energie do 80 ms a po 80 ms

$$C_{80} = 10 \log \frac{\int_0^{80ms} p^2(t) dt}{\int_{80ms}^{\infty} p^2(t) dt} = 10 \log \frac{E_{80}}{E_{\infty} - E_{80}}. \quad (1.21)$$

V současné době představuje zatím nejlepší korelaci poslechového atributu „jasnost“ pro časové rozlišení tónů v rychlých hudebních pasážích. Optimální hodnoty  $C_{80}$  silně záleží na hudebním žánru, obecně lze považovat za optimální hodnoty v rozmezí  $1 \pm 2$  dB. Pro romantickou hudbu by měli ležet v rozsahu hodnot  $-2$  dB až  $+4$  dB, zatímco pro klasickou a moderní hudbu je rozsah hodnot 0 dB až  $+6$  dB.

### 1.6.9 Zřetelnost $D_{50}$

Tato hodnota je definována poměrem energií časových úseků

$$D_{50} = C_{50} * 100 = \frac{E_{50}}{E_{80}}. \quad (1.22)$$

Hodnoty zřetelnosti jsou svou hodnotou ekvivalentem parametru *jasnost*.

### 1.6.10 Těžišťový čas $T_s$

Čas těžiště impulsové odezvy je úzce svázán s hodnotou  $C_{80}$  a má obdobné využití. Tato veličina má vypovídací hodnotu prostorového dojmu a jasnosti pro hudbu a mluvené slovo

$$T_s = \frac{\int_0^\infty t * p^2(t) dt}{\int_0^\infty p^2(t) dt} = \frac{\int_0^\infty t * p^2(t) dt}{E_\infty}. \quad (1.23)$$

Pro hudbu se hodnoty v oktávovém pásmu 1000 Hz pohybují v rozsahu 70 až 150 ms, pro řeč v oktávových pásmech 500 Hz až 4000 Hz jsou hodnoty v rozsahu 60 až 80 ms.

### 1.6.11 Míra doznívání $H$

Míra doznívání popisuje dozvukový prostorový dojem hudebního výkonu. Vypočítá se pro oktávové pásmo 1000 Hz z dekadického logaritmu poměru zvukové energie od 50 ms, ku energii přicházející v čase do 50 ms.

### 1.6.12 Interaurální koeficient křížové korelace $IACC$

Rozdíl mezi signály, které dorazí k levému a k pravému uchu. Souvisí se subjektivní kvalitou „dojmu prostorovosti“ v koncertním sále. Hodnoty  $IACC_E$  by se měly podle nejprísnějších požadavků v oktávových pásmech 500, 1000 a 2000 Hz pohybovat v rozsahu hodnot 0,4 až 1, kde hodnota 1 znamená maximální rozdílnost zvukových signálů v obou uších, tzn. zvukové pole je maximálně difúzní.

### 1.6.13 Echo Criterion $EK$

Dalším důležitým parametrem akustické kvality prostoru souvisejícím s odrazy je přítomnost, či nepřítomnost vnímatelné nebo dokonce rušivé ozvěny. První práce publikoval počátkem 50. let Helmuth Haas [16]. Na vzorku 80 osob, kterým přehrával pomocí dvou reproduktorů řečový signál, sledoval procento posluchačů rušených zpožděným signálem stejné úrovně v závislosti na časovém odstupu a pro 3 různé kadence řeči. V jiném experimentu zkoumal rušivý účinek zpožděného odrazu v závislosti nejen na časovém odstupu, ale i na jeho relativní úrovni. Při zpoždění o 10 až 30 ms musí stejně hlasitě slyšitelný odraz mít o 10 dB vyšší úroveň. Tento výsledek bývá v literatuře označován „Haasův jev“. V této práci byl pro měření zvolen  $\Delta\tau_E = 14$  ms [17].

## 2 IMPLEMENTACE A MĚŘENÍ ALGORITMŮ V MATLABU

Prostředí programu *Matlab* nabízí velkorysé možnosti při návrhu, simulacích a testování algoritmů. K signálu je během zpracování snadný přístup v podstatě kdekoli na jeho cestě a užitečné jsou i prefabrikované funkce. Výpočet dozvuku algoritmu implementovaného ve skriptu *Matlabu* mimo reálný čas spočívá v načtení celého vstupního signálu a provedení komplexní filtrace a zpoždění nebo vytvoření impulzní odezvy a její konvoluce se vstupním signálem vzorek po vzorku. Tento způsob implementace nemůže fungovat v reálném čase, ale slouží jako nejjednodušší způsob zjištění vlastností daných algoritmů.

V této práci jsou algoritmy napsané v programu *Math Works Matlab verze R2016a 9.0.0.341460*. Algoritmy jsou implementované tak, aby co nejčastěji využívali volání externích funkcí, ve kterých může implementovaný například fázovací článek. Tento způsob strukturace usnadňuje manipulaci s kódem pro experimentování s počtem nebo posloupností funkčních bloků za účelem zlepšení akustických vlastností dozvuku. Pro zvýšení praktičnosti a univerzálnosti lze na vstup přivést signál s libovolným vzorkovacím kmitočtem. Stejně tak lze na vstup přivést jak monofonní, tak stereofonní signál. Pro použití na reálných zvukových signálech je užitečná funkce *dry/wet*, která nastavuje poměr mezi zpracovaným a nezpracovaným signálem. V rámci testování byl na vstupu jednotkový impulz a signál ve větvi s nezpracovaným výstupem byl nulový.

Nejčastěji používanou funkcí byla funkce *filter*, která má jako vstupní parametry čítatel a jmenovatel přenosové funkce  $H(z)$  požadovaného filtru a vstupní signál. Tvar přenosové funkce určuje typ filtrace neboli způsob úpravy frekvenčního spektra. Nejjednodušší číslicový filtr je zpoždění s přenosovou funkcí

$$H(z) = z^{-k}, \quad (2.1)$$

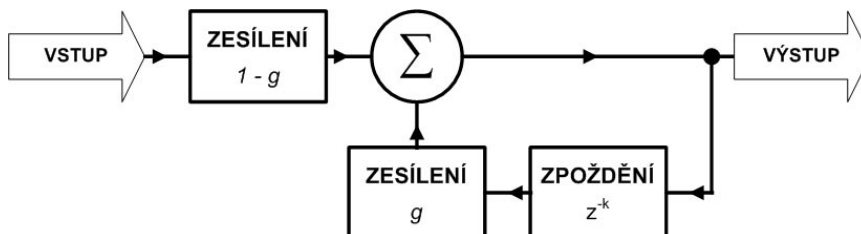
kde  $k$  je počet zpožděných vzorků. Dalším filtrem je dolní propust s přenosovou funkcí

$$H(z) = \frac{1 - g}{1 - gz^{-1}}, \quad (2.2)$$

kde  $g$  značí zesílení filtru prvního řádu vypočítaného ze vzorce

$$g = 2 - \cos\left(2\pi \frac{f_c}{f_s}\right) - \sqrt{\left(\cos\left(2\pi \frac{f_c}{f_s}\right) - 2\right)^2 - 1}, \quad (2.3)$$

který počítá se zadaným mezním a vzorkovacím kmitočtem,  $f_c$  resp.  $f_s$ . Na obrázku 2.1 je schéma tohoto filtru.

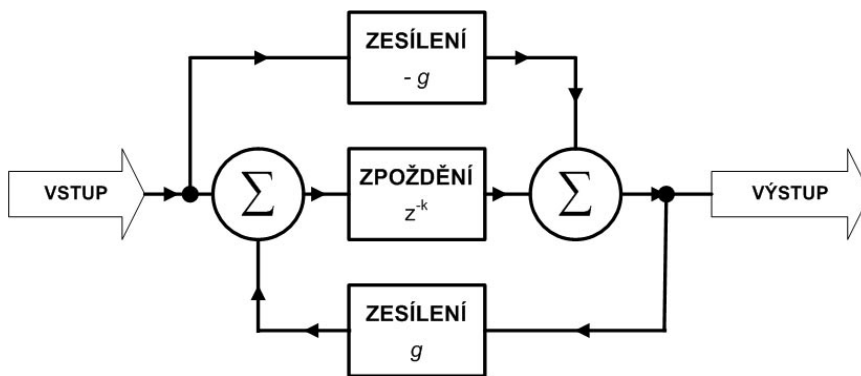


Obr. 2.1: Schéma filtru typu dolní propust.

Fázovací článek, jehož schéma je na obrázku 2.2 je klíčový u Dattorrova a Gardnerova algoritmu. Využívá se jeho vlastnosti, kdy při průchodu signálem zůstane neovlivněné frekvenční spektrum, ale změní se fáze signálu. Jeho přenosová funkce

$$H(z) = \frac{g + z^{-k}}{1 + gz^{-k}} \quad (2.4)$$

má zesílení  $g$  nastavené prostřednictvím zadaného vstupního řídicího parametru [7].



Obr. 2.2: Schéma fázovacího článku.

## 2.1 Implementace algoritmu Dattorrova reverberátoru

Vstupní parametry pocházejí z doporučení, které Dattorro uvedl v [6] na základě svého empirického výzkumu. Pro tuto implementaci se liší hodnota *vstupní difúze*, která byla původně 0,7. Změna na 0,95 přinesla zlepšení tvaru impulzní odezvy, přestože Dattorro doporučuje nastavovat hodnoty parametrů *diffusion* kolem 0,5. Celkový výčet vstupních parametrů je v tabulce A.1. Doporučený rozsah parametrů ovlivňujících difúzi je od 0 do 0,999. Pokud hodnota parametru překročí hodnotu 1, může vzniknout nestabilita. Změna hodnot na záporné je možná, změní se tak charakter impulzní odezvy, ale nebude porušena přenosová funkce fázovacího článku [6]. Mimoto krajní hodnoty některých parametrů způsobují chyby v chování algoritmu. Všechny koeficienty zpoždění byly nastavené pro vzorkovací kmitočet 29,761 kHz, ale zároveň dochází k přepočtu těchto hodnot v závislosti na vzorkovacím kmitočtu vstupního signálu. Nezbytný údaj potřebný k vytvoření vjemu prostoru ve výstupním signálu je tabulka koeficientů zpoždění výstupů, jejíž použití zaručí na výstupu uměle vytvořený stereofonní signál, přestože bude na vstupu signál pouze monofonní. Hodnoty a jejich polarita je uvedena v následující tabulce.

Tab. 2.1: Seznam hodnot sloužících k vytvoření stereofonního výstupu.

<b>výstupní body levého kanálu</b>	<b>zpoždění</b>	<b>polarita</b>
delay_left_n4	266	+
delay_left_n44	2974	+
delay_left_n5	1913	-
delay_left_n6	1996	+
delay_left_n1	1990	-
delay_left_n2	187	-
delay_left_n3	1066	-
<b>výstupní body pravého kanálu</b>	<b>zpoždění</b>	<b>polarita</b>
delay_right_n1	353	+
delay_right_n11	3627	+
delay_right_n2	1228	-
delay_right_n3	2673	+
delay_right_n4	2111	-
delay_right_n5	335	-
delay_right_n6	121	-



Ve skriptu se jako první operace provádí získání přenosových funkcí všech funkčních bloků, které slouží k výpočtu přenosové funkce *dekorelačního úseku* a struktury *tank*. Níže je uvedený kód pro vytvoření přenosové funkce pro fázovací článek *dekorelačního úseku*:

```
% čitatel přenosové funkce
num = [gain [zeros(1,delay-1) 1]];
% jmenovatel přenosové funkce
den = [1 [zeros(1,delay-1) gain]];
% vytvoření přenosové funkce
H_allpass04 = tf(num,den,Ts);
```

Přenosová funkce *dekorelačního úseku* a jednotlivých větví *tanku* vznikne prostým vynásobením patričních přenosových funkcí získaných během předchozí operace. Následně je vypočtena přenosová funkce celé struktury *tank*, a to podle vztahu

$$T(z) = \frac{vstup}{referenční\ signál} = \frac{1 + H_2(z)}{1 - H_1(z)H_2(z)}. \quad (2.5)$$

Výpočet  $T(z)$  je výpočetně nejnáročnější operací celého algoritmu.

Výstupy se tvoří v určitých bodech ve struktuře *tank*. Níže se nachází část kódu implementující jeden z výstupů. Tento výstup odpovídá výstupu znázorněnému zelenou šipkou na schématu na obr. 1.4.

```
% čitatel a jmenovatel přenosové funkce filtru
num = [decaydiffusion02 [zeros(1,delay6_2-1) 1]];
den = [1 [zeros(1,delay6_2-1) decaydiffusion02]];
% filtrace signálu ve smyčce tank
tank_signal_9 = filter(num,den,tank_signal_8);

% výpočet pro levý kanál

num = [zeros(1,delay_left_n6-1) 1];
% filtrace a uložení do pomocné proměnné
storage_left = filter(num,1,tank_signal_9);
% součet výstupu levého kanálu LP filtru z předchozího bodu a výstup AP uloženého v pomocné proměnné storage_left
output_left = summator(output_left,storage_left);

% výpočet pro pravý kanál

num = [zeros(1,delay_right_n6-1) 1];
% filtrace vstupu s opačnou polaritou a uložení do pomocné proměnné
storage_right = filter(num,1,-tank_signal_9);
% součet výstupu pravého kanálu LP filtru z předchozího bodu a výstup AP uloženého v pomocné proměnné storage_right
output_right = summator(output_right,storage_right);
```

Tyto výstupy se nakonec sečtou ve výslednou matici, která zahrnuje kromě stereo-fonního výstupu také počáteční zpoždění a prvotní odrazy:

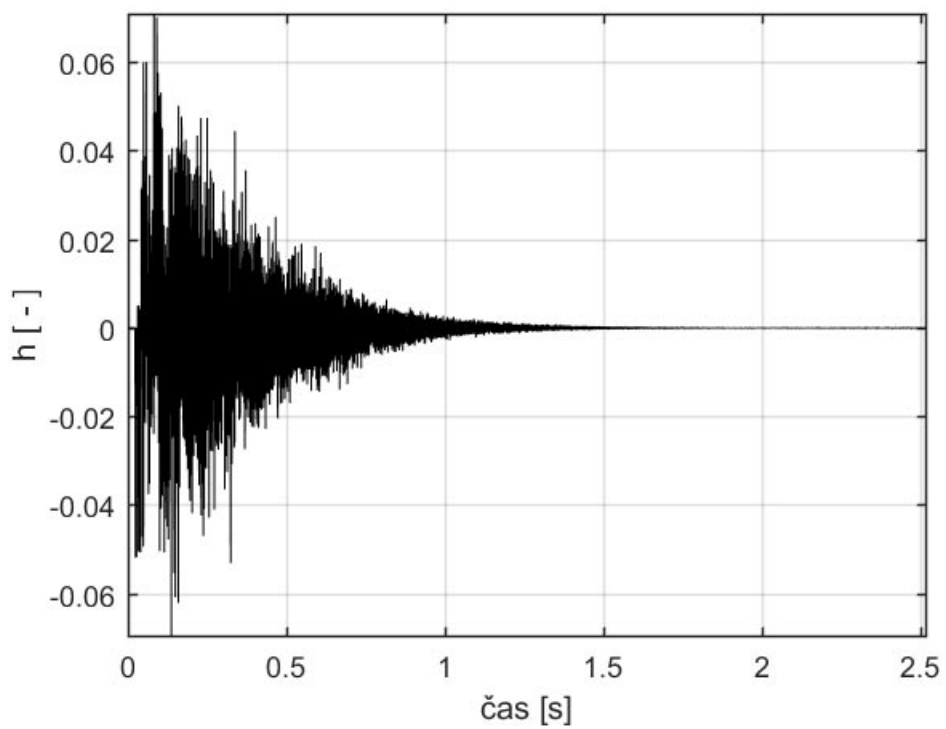
```
output_stereo_wet = ...
[zeros(1,predelay_samples) output_early_reflections output_left;
zeros(1,predelay_samples) output_early_reflections output_right];
```

Následující tabulka obsahuje změřené objektivní akustické parametry sestavy s nastavením, které navrhl Dattorro ve své práci:

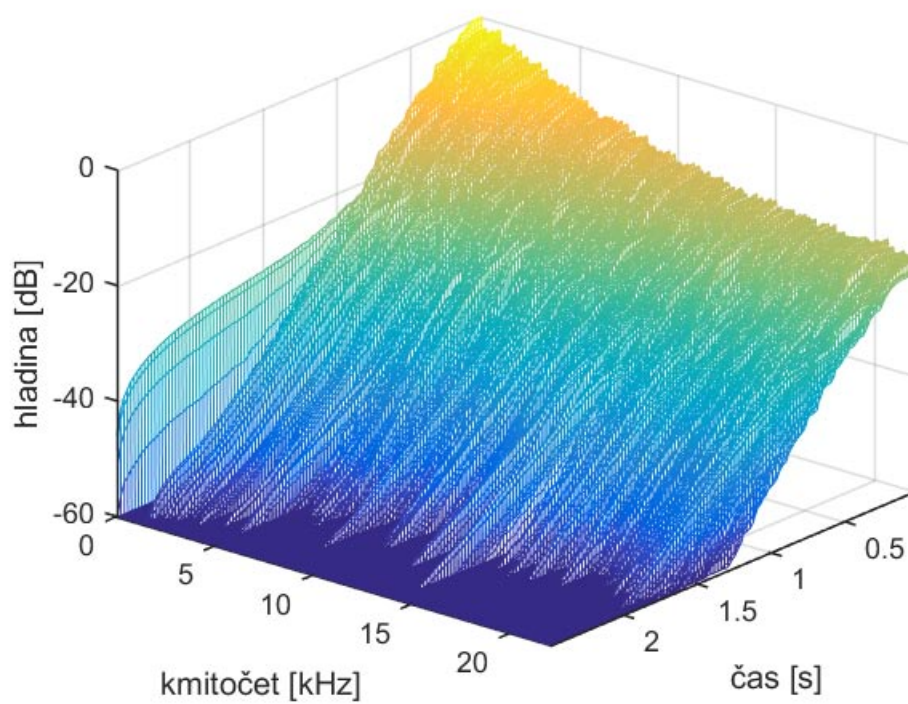
Tab. 2.2: Objektivní akustické parametry Dattorova reverberátoru.

<b>název akustické veličiny</b>	<b>zkratka</b>	<b>jednotka</b>	<b>hodnota</b>
doba dozvuku	$RT$	s	1,470
počáteční doba dozvuku	$EDT$	ms	1,991
míra přímého zvuku	$C_7$	dB	-47,694
jasnost	$C_{50}$	dB	-21,169
míra jasnosti	$C_{80}$	dB	-12,262
zřetelnost	$D_{50}$	%	0,758
míra doznívání	$H$	dB	21,169
počáteční prodleva	$ITDG$	ms	16,168
míra hlubokých tónů	$BR$	-	1,005
míra vysokých	$HFR$	-	1,000
těžiškový čas	$T_s$	ms	185,506
kritérium ozvěny	$EK_{max}$	-	0,891
interaaurální koeficient křížové korelace	$IACC_E$	-	0,029

Na obr. 2.3 je zobrazení impulzní odezvy. Vykazuje požadovaný exponenciálně klesající průběh obálky. Nárůst hustoty odrazů v tomto případě není dostatečný, ale před úpravou *vstupní difúze 1* byl tento nárůst ještě pomalejší. Impulzní odezva při poslechu také vykazovala známky tonálního zabarvení a kovový charakter dozvuku. Na tento fakt ovšem Dattorro dopředu upozornil v [6]. Reliéf poklesu energie (EDR) na obr. 2.4 je 3D graf, který zobrazuje křivku poklesu zvukové energie (EDC) v závislosti na kmitočtu. Všechny osy grafu jsou v lineárním měřítku, ačkoliv pro zobrazení kmitočtové osy se obvykle používá logaritmické měřítko. Z grafu je patrné, na kterých kmitočtech se objevují rezonanční složky. Výrazná složka na kmitočtu 0 je chyba, která vznikne, pokud se měří EDR z vygenerované impulzní odezvy ve zvukovém souboru, namísto přímo ve skriptu algoritmu.



Obr. 2.3: Impulzní odezva Dattorrova reverberátoru.



Obr. 2.4: Reliéf poklesu energie Dattorrova reverberátoru.

## 2.2 Implementace algoritmu Gardnerova reverberátoru

Implementace Gardnerova algoritmu vyžaduje odlišný přístup, než v předešlém případě. Výpočet přenosové funkce pro jednotlivé funkční bloky probíhá odděleně, zvlášť pro čitatele a jmenovatele, jak je popsáno v rovnici 1.5. V textu dále se objevují zkratky *SNAP* a *DNAP*, které znamenají *single nested allpass* a *double nested allpass*. Jedná se o jednoduchý vnořený fázovací členek a dvojitý vnořený fázovací členek, který se od jednoduchého liší tím, že uvnitř struktury jednoho fázovacího článků jsou vloženy dva sériově zapojené fázovací články.

Tento algoritmus obsahuje ve své struktuře i modul tvořící prvotní odrazy. Jejich amplituda je nastavená tak, aby korespondovala s difúzní částí dozvuku. Výstup tohoto modulu je přiveden na výstup a zároveň na vstup recirkulační zpožďovací smyčky. Implementace prvotních odrazů pomocí FIR filtru je poněkud zjednodušující, a tak kvalita modelování prvotních odrazů místnosti není zcela dostačující. Obzvláště u reverberátorů typu *room* tvoří prvotní odrazy kritickou část impulzní odezvy a posluchačovi dávají na rozdíl od difúzního dozvuku jasnější informaci o tvaru a velikosti prostoru. Ve skriptu je provedena úprava, která z počáteční části difúzní části dozvuku odebírá množství vzorků odpovídající přibližně 70 ms. Je to z toho důvodu, že algoritmus vykazuje z neznámého důvodu nezvykle velkou počáteční prodlevu. Tabulka s kompletním seznamem vstupních parametrů je součástí přílohy A.2. Vstupní parametry jsou zvoleny na základě doporučení z [10].

Výpočty čitatele a jmenovatele přenosové funkce probíhají u všech bloků recirkulační smyčky v pořadí, které je určeno návrhem topologie. Externí funkce vrací spočítané čitatele a jmenovatele odděleně v různých proměnných. Funkce se liší v závislosti na tom, pro jaký typ filtru je upraven způsob výpočtu. Strukturace kódu algoritmu a vyčlenění jednoduchých úloh jako externí funkce zjednodušuje případné následné úpravy a rozšíření. Tento způsob navíc zvyšuje přehlednost hlavního skriptu. Výsledky pro jeden blok, například fázovací členek, se následně uloží proměnných *num\_loop* a *den\_loop*. Při výpočtu dalšího bloku se provede konvoluce výsledků z tohoto bloku s proměnnými *num\_loop* a *den\_loop*, které obsahují výsledky všech předchozích bloků.

Na konci celé smyčky tak obsahují proměnné *num\_loop* a *den\_loop* informaci o celkové přenosové funkci, která je dále využita pro vytvoření signálu na výstupu. Následující část kódu zobrazuje příklad výpočtu čitatele a jmenovatele přenosové funkce pro jednoduchý vnořený fázovací členek (SNAP). Následně probíhá konvoluce s proměnnými *num\_loop* a *den\_loop*, které již obsahují čitatele a jmenovatele přenosových funkcí fázovacího článku 1 a 2, a zpoždění 1 a 2.

```

% čitatel vnějšího vnořeného AP
num_i = [gain_nested_inner [zeros (1,delay_n1_inner - 1) 1]];
% čitatel rozdílu pro výpočet num
num_oi = [zeros(1,(delay_n1_outer - delay_n1_inner) - 1) 1];
% jmenovatel vnějšího vnořeného AP
den_i = [1 [zeros(1,delay_n1_inner - 1) gain_nested_inner]];
num = conv(num_i,num_oi);
num = summator(- gain_nested_outer * den_i,num); % čitatel SNAP
num_d = conv(num_i,num);
den = summator(den_i,-gain_nested_outer * num_d); % jmenovatel SNAP

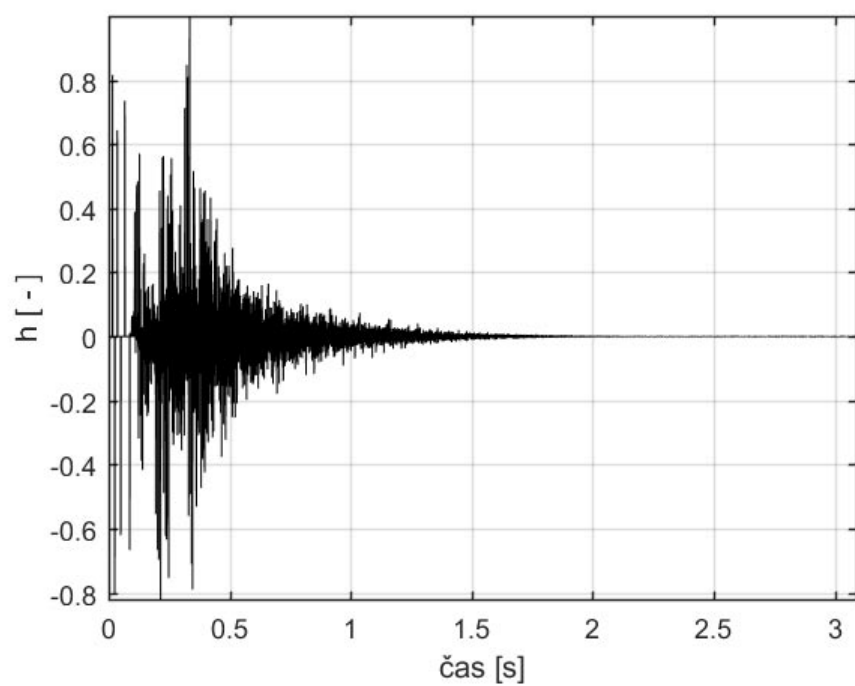
num_loop = conv(num_loop,num);
den_loop = conv(den_loop,den);

```

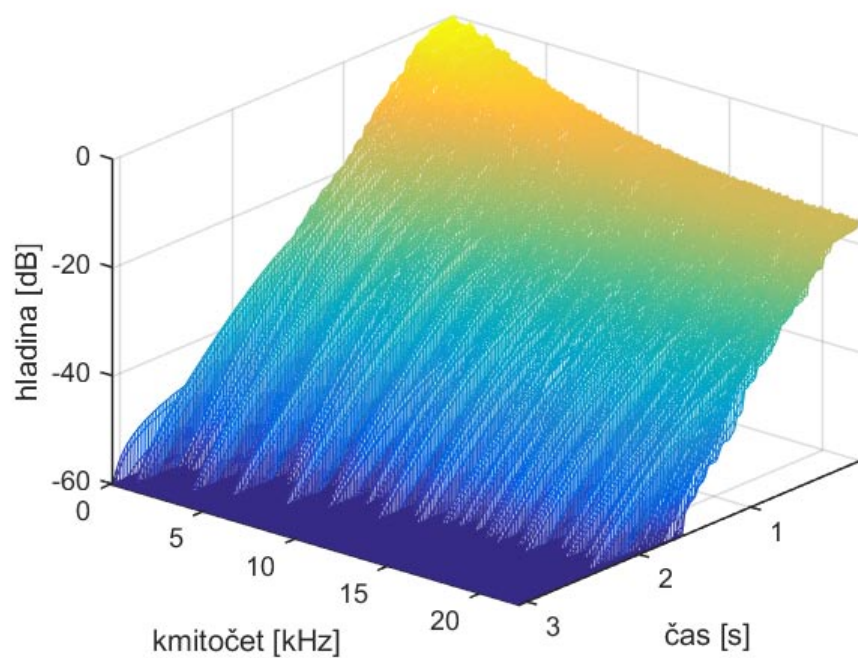
Výstupy jsou tvořené na třech místech, jak je vidět na obr. 1.7. Výstup číslo 1 nejprve filtruje vstupní signál funkcí *filter* s koeficienty přenosové funkce *num\_loop* a *den\_loop* vytvořených v předchozím kroku. Následně jsou zavolané funkce realizující fázovací článek a zpoždění. Výstup je váhovaný koeficientem zesílení, který Gardner stanovil na 0,34. Výstup číslo 2 má na svém vstupu signál z výstupu číslo 1 a následně volá funkce pro zpoždění a SNAP. Výstup číslo 2 je váhovaný odlišným koeficientem zesílení a navíc sečtený s výstupem číslo 1. Třetí výstup je vytvořený stejným postupem.

Tab. 2.3: Objektivní akustické parametry Gardnerova reverberátoru.

název akustické veličiny	zkratka	jednotka	hodnota
doba dozvuku	$RT$	s	2,44
počáteční doba dozvuku	$EDT$	ms	3,29
míra přímého zvuku	$C_7$	dB	-52,14
jasnost	$C_{50}$	dB	-38,87
míra jasnosti	$C_{80}$	dB	-28,46
zřetelnost	$D_{50}$	%	0,01
míra doznívání	$H$	dB	38,87
počáteční prodleva	$ITDG$	ms	15,22
míra hlubokých tónů	$BR$	-	1,001
míra vysokých	$HFR$	-	0,98
těžišťový čas	$T_s$	ms	318,24
kritérium ozvěny	$EK_{max}$	-	2,00



Obr. 2.5: Impulzní odezva Gardnerova reverberátoru.



Obr. 2.6: Reliéf poklesu energie Gardnerova reverberátoru.

## 2.3 Implementace algoritmu Jotova reverberátoru

Jotův algoritmus má oproti předešlým odlišně pojatý přístup k získání impulzní odezvy. Implementace v *Matlabu* totiž neumožňuje použít funkci *filter*, která by dokázala pracovat na vstupech s dvourozměrnými maticemi. To vede k použití cyklu *for*, a nyní je impulzní odezva na základě vstupních parametrů vypočítaná vzorek po vzorku[7]. Tento způsob není příliš efektivní, ale v prostředí *Matlabu* jediný možný. Výstupní monofonní signál tohoto reverberátoru je získaný konvolucí impulzní odezvy se vstupním signálem.

Přesto ve srovnání s topologií navrženou Dattorrem či Gardnerem je výpočet pro stejnou dobu dozvuku znatelně menší. Během testování vyšlo najevo, že zvýšení hodnoty vstupního parametru *doba dozvuku* téměř exponenciálně zvyšuje čas pro výpočet jednoho vzorku. Výpočet dozvuku pro delší signály je taktéž časově náročný. Tabulka A.3 obsahuje seznam všech vstupních parametrů. Hodnoty koeficientů zesílení na vstupu a výstupu všech čtyř zpožďovacích linek zůstávají po dobu měření na výchozí hodnotě 1, která znamená, že signál není ovlivněn. Jejich nastavení pro získání požadovaného tvaru impulzní odezvy je poněkud zdlouhavé. Přesto bylo vyzorováno, že změna parametru  $b_i$  ovlivňuje výsledky měření objektivních akustických parametrů více, než zesílení  $c_i$  na výstupu. Ve zpětné vazbě byla použita tato matice 1.13. Koeficienty zpoždění  $z^{-d}$  by měli být vybrané tak, aby daná čísla byla prvočísla a poměr mezi nejmenší a největší hodnotou byl maximálně 1 : 2, 5. Dobu dozvuku ovlivňuje parametr *imp\_length* tím, že určuje počet opakování cyklu generujícího impulzní odezvu. Parametr *imp\_length* je určen součinem parametru *reverb\_time* a vzorkovacího kmitočtu. Koeficienty zpětnovazební matice jsou na parametru *reverb\_time* nezávislé.

Následuje úsek kódu Jotova algoritmu, kde probíhá výpočet impulzní odezvy:

```
for i = 1 : imp_length

    % jednotlivé zpožďovací linky jsou násobené absorpčním zesílením
    buffer(:,i) = buffer(:,i) .* gainmatrix;
    % výstup obsahující jednotlivé impulzní odezvy
    imp_fract(:,i) = buffer(:,i);
    % násobení maticí A a načtení do zpětné vazby
    feedback = buffer(:,i)' * A;
    % přičtení zpětnovazebního signálu
    buffer(1,i+delay(1)) = feedback(1);
    buffer(2,i+delay(2)) = feedback(2);
    buffer(3,i+delay(3)) = feedback(3);
    buffer(4,i+delay(4)) = feedback(4);
end
```

```

% zesílení c
imp_fract(1,:) = imp_fract(1,:) .* c1;
imp_fract(2,:) = imp_fract(2,:) .* c2;
imp_fract(3,:) = imp_fract(3,:) .* c3;
imp_fract(4,:) = imp_fract(4,:) .* c4;

% výpočet impulzní odezvy
imp_resp = sum(imp_fract);

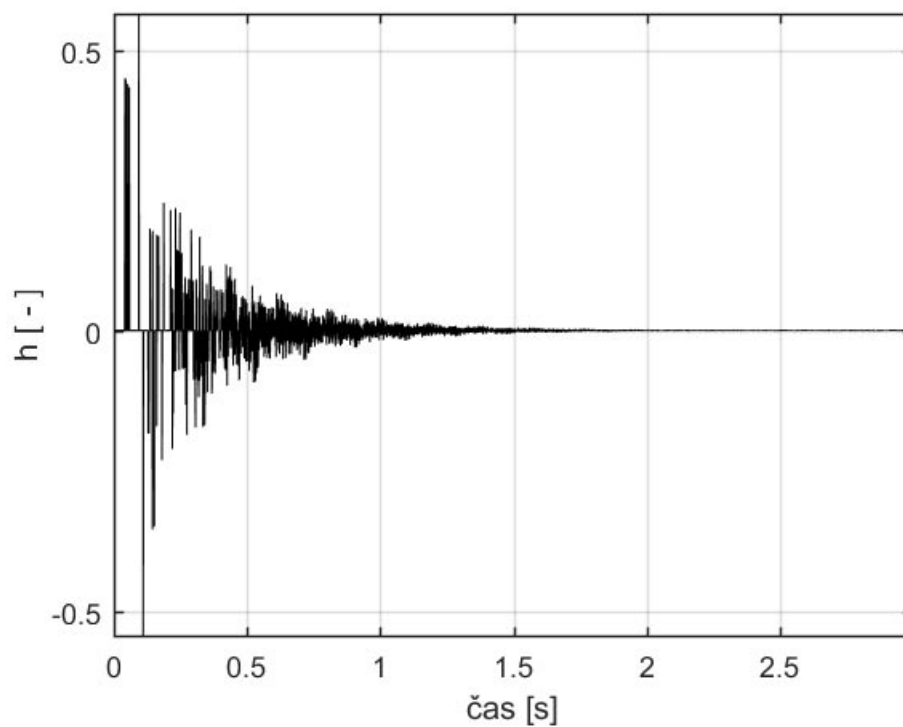
```

Tabulka 2.4 obsahuje změřené objektivní akustické parametry Jotova reverberátoru. Měření probíhalo při nastavených parametrech, jejichž hodnoty jsou v tabulce A.3. Při stejných parametrech byl změřen i průběh impulzní odezvy a reliéf poklesu energie, které jsou k nalezení na další stránce.

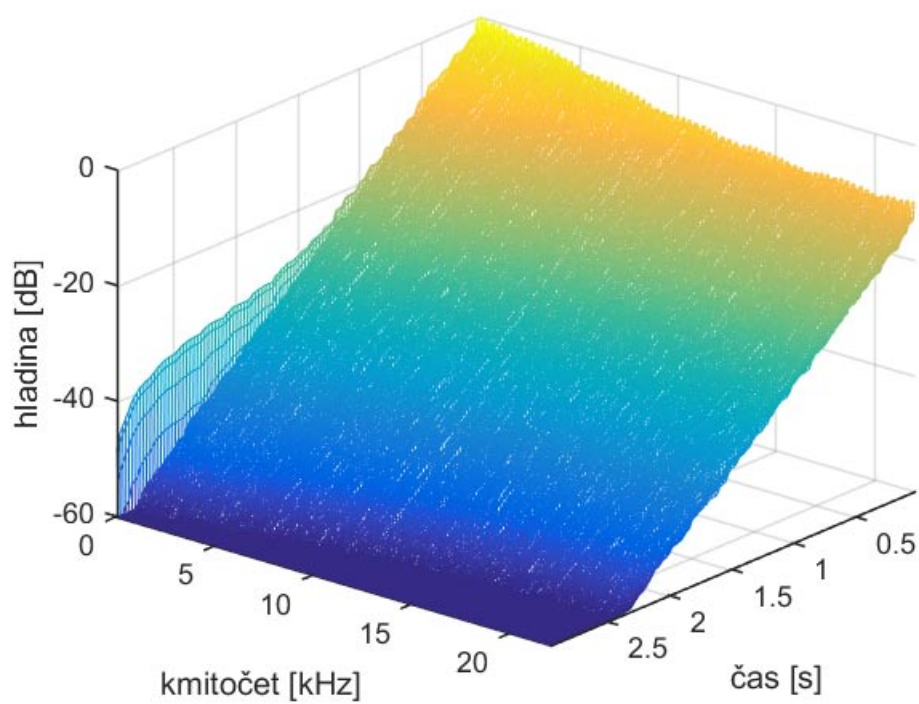
Tab. 2.4: Objektivní akustické parametry Jotova reverberátoru.

název akustické veličiny	zkratka	jednotka	hodnota
doba dozvuku	$RT$	s	1,51
počáteční doba dozvuku	$EDT$	ms	1,32
míra přímého zvuku	$C_7$	dB	-14,05
jasnost	$C_{50}$	dB	-4,83
míra jasnosti	$C_{80}$	dB	-4,95
zřetelnost	$D_{50}$	%	24,73
míra doznívání	$H$	dB	4,83
počáteční prodleva	$ITDG$	ms	41,88
míra hlubokých tónů	$BR$	-	1,04
míra vysokých	$HFR$	-	1,00
těžišový čas	$T_s$	ms	131,12
kritérium ozvěny	$EK_{max}$	-	0,80





Obr. 2.7: Impulzní odezva Jotova reverberátoru.



Obr. 2.8: Reliéf poklesu energie Jotova reverberátoru.

## 3 IMPLEMENTACE JOTOVA REVERBERÁTORU POMOCÍ TECHNOLOGIE VST

### 3.1 Vývojové prostředí

*Audio System Toolbox* je rozšíření programu *Matlab* nabízené od verze 2016a, jehož hlavní předností je možnost implementace VST zásuvného modulu z kódu *Matlabu*. Rozšíření vzniklo na základě faktu, že mnozí vývojáři při vytváření nových VST zásuvných modulů často své algoritmy testují v *Matlabu* před samotnou implementací algoritmu v jazyce *C++*. Umožňuje tak zefektivnění pracovního postupu tím, že vytváří mezikrok na cestě od návrhu v podobě algoritmu implementovaného v *Matlabu* k vytvoření plnohodnotného VST v objektově orientovaném programovacím prostředí. Tento mezikrok zrychluje tempo vývoje a testování, a usnadňuje odstranění chyb. Mezi další výhody patří interakce s MIDI zařízeními a možnost opětovného použití kódu. Na druhou na toto rozšíření nelze pohlížet jako na nástroj k vytvoření zásuvného modulu komerční úrovně.

Pro práci s rozšířením *Audio System Toolbox* (dále jen *AST*) je vhodná alespoň základní znalost principů objektově orientovaného programování. Při návrhu algoritmu ve třídě `audioPlugin` postačuje jen základní znalost OOP. Při návrhu s navíc zahrnutou třídou `matlab.System` je vyžadováno větší porozumění OOP v prostředí *Matlab* [18]. Výhoda je větší množství rozšiřujících funkcí a možnost integrace objektů do *Simulinku*.

K vygenerování VST zásuvného modulu je potřeba překladač třetí strany provádějící překlad z vstupního programovacího jazyka do jazyka výstupního. V operačním systému Windows jsou podporovány tyto překladače: *Microsoft Visual C++ 2013 Professional*, *Microsoft Visual C++ 2012 Professional* a *Microsoft Visual C++ 2010 Professional SP1*. Ve výchozím nastavení probíhá generování 64 bitových VST zásuvných modulů verze 2.4. Je možné generovat i 32 bitové verze. Vygenerované zásuvné moduly jsou kompatibilní s většinou 32 i 64 bitových verzí hostitelských aplikací fungujících v operačním systému *Windows*.

#### 3.1.1 Nástroj Audio Test Bench

*Audio Bench Test* je klíčový nástroj poskytující grafické prostředí, pomocí kterého lze vyvíjet, testovat, odstraňovat chyby a ladit algoritmus v reálném čase. Simuluje prostředí hostitelské aplikace a umožňuje vizualizovat zpracovávaný signál v časové i kmitočtové doméně. Na vstup je možné přivést zvukový signál ze souboru, či přímo z mikrofону. Výstup může být přehraný v reálném čase, zapsaný do souboru nebo

obě možnosti zároveň. Ovladač zvukového zařízení je ve výchozím nastavení nastavený na DirectSound, ale lze přepnout i na ovladač ASIO.

### 3.1.2 Doporučený pracovní postup

Doporučený obecný pracovní postup se skládá z těchto kroků:

1. Vytvoření algoritmu zásuvného modulu v *Matlabu* na základě platné syntaxe.
2. Testování na programátorské chyby a na chyby v algoritmu.
3. Dynamické testování pomocí příkazu `validateAudioPlugin` slouží k vyhledání chyb, které by statické testování probíhající v rámci `generateAudioPlugin` neodhalilo.
4. Testování nástrojem *Audio Test Bench* pro zjištění vlastností a chování vytvořeného algoritmu poslechovým testem i vizualizací výstupu v časové i kmitočtové rovině v reálném čase.
5. Zadání příkazu `generateAudioPlugin` pro vygenerování VST zásuvného modulu v podobě dynamicky připojované knihovny s příponou *.dll*.
6. Testování v prostředí hostitelské aplikace.

### 3.1.3 Třída `AudioPlugin`

Převzaté označení *audio plugin* odkazuje na obecné označení třídy odvozené od základní třídy `audioPlugin`, která umožňuje přístup k funkcím rozšíření *AST*. Algoritmy jsou vytvořené ve třídě *audio plugin* a název této třídy posléze odpovídá názvu zásuvného modulu. Ve třídě `audioPlugin` lze vytvářet zvukové efekty, ve třídě `audioPluginSource` je možné vytvořit generátory zvukového signálu. Dvě základní podmínky pro vytvoření třídy *audio plugin* udávají, že třída *audio plugin* musí dědit vlastnosti třídy `audioPlugin`, a že musí obsahovat funkci pojmenovanou *process* [18].

Na následující stránce je kód představující základní šablonu, ve které lze vytvořit zásuvný modul. Popis a vysvětlení funkce jednotlivých sekcí je popsán v komentářích přímo v kódu.

```

classdef Nazev_zasuvneho_modulu < audioPlugin

    % Nazev_zasuvneho_modulu je konkrétní označení třídy audio plugin
    % Třída Nazev_zasuvneho_modulu dědí vlastnosti třídy audioPlugin

    properties

        % V této sekci probíhá inicializace veřejných vlastností, které
        % jsou přístupné uživateli.
    end
    properties (Access = private)

        % Tato sekce slouží k inicializaci soukromých vlastností, ke kterým
        % nemá koncový uživatel přístup, např. stavové proměnné.
        % Umožňuje ukládat informace do objektu.
    end
    properties (Constant)

        % Zde se definuje, jaké ovládací prvky se objeví uživateli, včetně
        % nastavení rozsahu možných hodnot a pojmenování parametru.
    end
    methods

        function out = process (plugin, in)

            % Tato sekce obsahuje algoritmus ke zpracování vstupního zvukového
            % signálu. Funkce musí být pojmenovaná process a její přítomnost
            % je nezbytná pro fungování algoritmu.
        end
        function reset (plugin)

            % Tato sekce obsahuje instrukce k aktualizování vlastností a
            % vyprázdnění stavových proměnných při spuštění a při změně
            % vzorkovacího kmitočtu hostitelské aplikace.
        end
        function set.MyProperty (plugin, val)

            % Zde jsou obsaženy příkazy k přepočítání hodnoty soukromých
            % vlastností, například při změně vzorkovacího kmitočtu.
            % Veřejné vlastnosti propojené s ovládacími prvky jsou
            % aktualizované automaticky.
        end
    end
end
end

```

## 3.2 Návrh algoritmu

Implementací algoritmu pomocí technologie VST získáme prostředí, ve kterém je možné zkoumat a testovat vlastnosti vytvořeného algoritmu v reálném čase. Pro tuto úlohu byla vybrána Jotova topologie reverberátoru, protože na rozdíl od Dattorovy a Gardnerovy je univerzálnější, má lepší zvukové vlastnosti a má výhodnější možnosti rozšíření.

Prvotní pokusy o vytvoření algoritmu a zjišťování způsobů transformace kódu tak, aby původní algoritmus byl co nejjednodušší cestou upraven do definované struktury, kterou *Audio System Toolbox* používá ke generování VST zásuvných modulů, nedopadly podle očekávání. Pokud by se kód transformoval tak, aby pracoval stejným principem, v rámci algoritmu by se dopředu vypočítala impulzní odezva na základě vstupních parametrů a následně by probíhala v reálném čase konvoluce s bloky vstupního signálu. Zatímco tento přístup byl pro implementaci Jotova algoritmu v prostředí *Matlabu* i přes jeho neefektivitu jediný možný, pro zpracování v reálném čase je nevhodný. Opuštění původního přístupu znamená vytvoření zásuvného modulu od úplného základu, namísto transformace původního kódu.

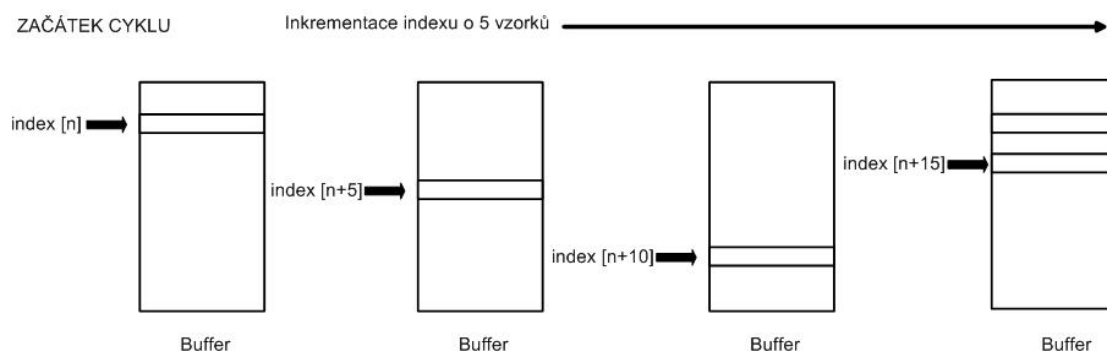
Při návrhu algoritmu je nyní třeba vzít do úvahy, že algoritmus nebude mít dopředu před započítím zpracování k dispozici celý vstupní signál. Na vstup nyní bude přicházet sled bloků zvukových dat s dopředu neznámou velikostí, např. 512 nebo 2048 vzorků. Tyto bloky se musí postupně v co nejkratším čase zpracovat a odeslat na výstup. Zpoždění mezi vstupem a výstupem systému vzniklé nenulovou dobou zpracování signálu se nazývá latence. O správu vstupů a výstupů se stará hostitelská aplikace. Ta také určuje velikost vstupních bloků a vzorkovací kmitočet. Tím, že algoritmus nemá přístup k celému signálu, vzniká potřeba ukládání dat ze vstupních bloků pro použití v dalších volání funkce *process*.

Situaci lze znázornit na příkladu. Na vstup systému přichází zvuková informace, např. jednotkový impulz v bloku o délce 512 vzorků. To znamená, že při jednom zavolání funkce *process*, algoritmus dostane ke zpracování 512 vzorků. Tento blok má při vzorkovacím kmitočtu 44100 Hz trvání 11,6 ms. Chceme-li vytvořit zpoždění 0,1 sekundy, které odpovídá při vzorkovacím kmitočtu 44100 Hz 4410 vzorkům, budeme muset uchovat data z prvního bloku a udržet je v paměti do doby, než bude zpracováván 9. blok. Zároveň se data z prvního bloku nesmí přepsat daty z bloků 2 až 8. Tento požadavek splňuje datová struktura, která se nazývá kruhová vyrovnávací paměť.

### 3.2.1 Kruhová vyrovnávací paměť

Vytvoření zpožďovacích linek, které pracují v reálném čase, vyžaduje implementaci datové struktury, která se nazývá kruhová vyrovnávací paměť. Výhoda spočívá mimo jiné v tom, že po použití prvku dané matice se nemusí všechny ostatní prvky přesouvat o jednu pozici. Tím dochází ke snížení nároků na paměť počítače a zlepšení efektivity zpracování signálu. Ke správě kruhové vyrovnávací paměti slouží dva ukazatele. Jeden k určení pozice, kam se má zapisovat nový prvek a druhý na pozici, ze které má číst nejstarší vložený prvek.

V oblasti číslicových signálových procesorů umožňuje jednotka pro generování adres (AGU) vytvořit vyrovnávací paměť, která bude obsahovat kruhové adresování automaticky. V jazyce *C++* nebo v tomto případě v *Matlabu* je potřeba vytvořit tento mechanismus svépomocí. Vytvořená matice představující vyrovnávací paměť má lineární adresování. To znamená, že ukazatel bude inkrementovat nebo dekrementovat indexové pozice podle nastavení a postupně se posouvat na další pozici. Pokud se ukazatel dostane na pozici překračující velikost vyrovnávací paměti, dojde k chybě a pravděpodobně k selhání programu. Ošetření zajišťuje při dosažení poslední pozice ve vyrovnávací paměti navrácení ukazatele zpět na počáteční pozici [19]. Na obr. 3.1 je znázorněný princip funkce kruhové vyrovnávací paměti.



Obr. 3.1: Princip kruhové vyrovnávací paměti.

Pozice zápisu je v prvním kroku algoritmu na první pozici. Pozice pro čtení je vypočtena na základě rozdílu aktuální zápisové pozice a nadefinované velikosti zpoždění ve vzorcích. Následuje podmínka, která k pozici pro čtení přičte hodnotu celkové velikosti vyrovnávací paměti, pokud je pozice pro čtení menší nebo rovna nule. Pokud je pozice pro čtení vypočítaná jako součet aktuální zápisové pozice a nadefinované velikosti zpoždění, tak při nastaveném minimálním zpoždění vznikne maximální definované zpoždění a obráceně. Po provedení příkazů stěžejní části algoritmu následuje inkrementace zápisové i čtecí pozice kruhové vyrovnávací paměti a ošetření proti překročení maximální velikosti paměti. Před provedením druhého

kroku programu je zjištěna aktuální hodnota zápisové pozice která zajistí, že v dalším kroku budou vstupní data návazně umístěna za poslední pozici, kam byla zapsána v prvním kroku.

## 3.3 Implementace

### 3.3.1 Sekce *Vlastnosti*

*Vlastnosti* umožňují zásuvnému modulu ukládat informace napříč všemi sekcemi třídy. Vlastní algoritmus pracuje s objekty. Tyto objekty musí být deklarovány jako *vlastnosti*, aby mohly být použity ve více sekcích. K *veřejným vlastnostem* je neomezený přístup. Obsahují proměnné s nadefinovanou výchozí hodnotou a jsou závislé na nastavení parametrů ovládacích prvků. *Soukromé vlastnosti* označené atributem (`access = private`) mohou používat pouze metody stejné třídy. Jestli je přístup k proměnným nastavený na *soukromý*, proměnná není přístupná koncovému uživateli. Zde jsou deklarovány proměnné jen svým názvem nebo se zde deklarují stavové proměnné s přiřazenou nulovou maticí. Dalším atributem *vlastností* je *konstanta*. Ta se využívá, pokud je dopředu známe, že bude přiřazena pouze jediná hodnota. Zde jsou takto definované koeficienty zesílení a zpoždění, na základě kterých jsou generovány prvotní odrazy. V rámci sekce vlastnosti nelze provádět přípravné výpočty, které by se vztahovaly k jiným vlastnostem.

### 3.3.2 Grafické rozhraní

K vytvoření interaktivního rozhraní zásuvného modulu v podobě jednoduchých ovládacích prvků slouží objekt `PluginInterface` deklarovaný v sekci *vlastnosti*. Propojí veřejnou vlastnost s ovládacím prvkem, a tak může být její hodnota ovlivňována ze strany uživatele v reálném čase. Na níže uvedená část programu představuje implementaci ovládacích prvků, kde `audioPluginParameter` je název veřejné vlastnosti a `DisplayName` zobrazovaný název ovládacího prvku v DAW. `Mapping` je rozsah hodnot, jakých může ovládací prvek nabývat.

```
properties (Constant)
    PluginInterface = audioPluginInterface(...
        audioPluginParameter('public_delay1',...
            'DisplayName','delay 1',...
            'Mapping',{ 'int',10,5000}));
end
```

Lze přidat libovolné množství ovládacích prvků, ale není možný jakýkoliv zásah do grafické stránky zobrazení jako rozmístění, velikost nebo definice podoby ovládacího prvku. Tím je myšleno, že nejde například určit, zda se bude prvek zobrazovat

jako otočný nebo posuvný potenciometr. Samotné grafické rozhraní vygeneruje až hostitelská aplikace a obsahuje jen nejzákladnější informace a jednoduché rozmístění ovládacích prvků. Výchozí hodnota parametru ovládacího prvku je určena hodnotou proměnné ze sekce *Vlastnosti*, na kterou se odkazuje.

### 3.3.3 Funkce *konstruktor*

*Konstruktor* je speciální funkce, která vytváří instanci třídy. Provádí inicializaci proměnných a jednoduché výpočty. Je to efektivnější způsob oproti situaci, kdy všechny výpočty probíhají uvnitř funkce *process*. Následkem redundantních výpočtů je větší latence systému. *Konstruktor* přijímá vstupní argumenty, přiřazuje data uložená ve vlastnostech a vrací inicializovaný objekt. Aby *Matlab* rozeznal, že tato funkce slouží jako konstruktor, musí mít stejný název, jako třída *audio plugin*, resp. název zvukového zásuvného modulu. Jediný výstupní argument *konstruktoru* je objekt jím vytvořený.

### 3.3.4 Funkce *reset*

Funkce *reset* slouží k aktualizaci vlastností, které závisí na vzorkovacím kmitočtu a vyprázdnění obsahu stavových proměnných. Tato funkce se volá, když se změní vzorkovací kmitočet hostitelské aplikace. Použitím funkce *getSampleRate* lze získat aktuální vzorkovací kmitočet a na jeho základě přepočítat závislé vlastnosti. Funkce *reset* by se neměla používat pro nastavení hodnot vlastností propojených s parametry ovládacích prvků [20].

### 3.3.5 Funkce *set*

Funkce *set* se používá, pokud je žádoucí prostřednictvím interaktivního prostředí ovládat i *soukromé vlastnosti*. V tomto případě zpoždění označené jako *public delay1*. Veřejná vlastnost *public delay1* je propojená s parametrem ovládacího prvku, který na základě rozhodnutí uživatele může měnit jeho hodnotu. *Public delay1* je ale také propojené se soukromou vlastností *delay1*, jejíž hodnota se mění i závislosti na aktuálním vzorkovacím kmitočtu. Implementace je naznačena v kódu níže:

```
function set.public_delay1 (plugin,public_delay1)
    plugin.public_delay1 = public_delay1;
    plugin.delay1 = round(public_delay1 *...
        (getSampleRate(plugin)/plugin.init_Fs));
end
```



### 3.3.6 Funkce *process*

Tato funkce obsahuje jádro celého algoritmu, které provádí samotné zpracování signálu. Na vstupu této funkce jsou argumenty *plugin* a *in*. Argument *in* představuje blok vstupního signálu. Na začátku funkce proběhne výpočet aktuální velikosti vstupních bloků a také vytvoření nulové matice pro výstup, jejíž velikost odpovídá velikosti aktuálního bloku na vstupu. Následuje implementace prvotních odrazů, které jsou realizovány FIR filtrem převzatým z původní implementace Jotova algoritmu. Poté probíhá výpočet pozic pro čtení z kruhové vyrovnávací paměti. Tyto pozice jsou aktualizovány s každým voláním funkce *process*.

Smyčka **for** má počet iterací omezený aktuální velikostí bloku vstupních dat. V rámci smyčky probíhá načtení součtu vstupního signálu a zpětné vazby na zápisovou pozici v kruhové vyrovnávací paměti. Vstupní signál je v této fázi rozdělen do samostatných větví, které představují jednotlivé zpožďovací linky, a vynásoben koeficientem vstupního zesílení  $b_i$ . Následně je signál na pozicích představujících zpoždění vynásoben absorpčním zesílením  $k_i$  a vstupuje do větve zpětné vazby, kde probíhá další násobení s maticí  $\mathbf{A}$ . Ta je shodná s maticí použitou i v původní implementaci. Signál ze zpětné vazby je v dalším cyklu smyčky **for** přičten ke vstupnímu signálu. Výstup je před filtrem tonální korekce sečtený a vynásobený výstupním koeficientem zesílení jednotlivých větví  $c_i$ .

Před vytvořením výstupu je signál filtrován IIR filtrem prvního řádu, typu dolní propust. Tento filtr se nazývá filtr tonální korekce a slouží k úpravě spektra již sečtených výstupů zpožďovacích linek. Jedná se o totožný filtr, jaký byl použitý i v původním algoritmu. Pro výpočet přenosové funkce filtru byla vytvořena externí funkce *LPcoeff.m*. Výpočet probíhá jen při spuštění zásuvného modulu a změně mezního nebo vzorkovacího kmitočtu, a tím snižuje výpočetní nároky algoritmu [21]. Funkce *filter* musí pro zpracování v reálném čase obsahovat stavové proměnné, které udrží v paměti počáteční podmínky definující filtr pro zpracování v dalším bloku. Stavové proměnné jsou inicializovány ve funkci *reset* při spuštění zásuvného modulu a při změně vzorkovacího kmitočtu.

Výstup je tvořen vzorek po vzorku. Funkce *process* má v základním nastavení stereofonní vstup a stereofonní výstup. Pro tento algoritmus je vstup signálu sečten do monofonní stopy, zpracován, a na výstupu se kanál duplikuje tak, aby byla splněna podmínka konzistence vstupu a výstupu. V závěru smyčky probíhá inkrementace pozic pro zápis a čtení a také ošetření proti překročení velikosti kruhové vyrovnávací paměti.

Následuje výpis stěžejní části algoritmu obsahující pro zjednodušení pouze jednu zpožďovací linku, namísto čtyř (argument `p` je v tomto případě zkratkou pro `plugin`):

```
% načtení vstupního signálu a násobení vstupů linek zesílením b
% a přičtení zpětné vazby na vstup
p.buffer(writeindex,1) = ERmonoin(i,1)*p.b1 + p.feedback(1,1);

% absorpční zesílení a zpoždění
p.buffer(readindex1,1)=p.buffer(readindex1,1)*p.absorb_gain(1);

% načtení do zpětné vazby
p.feedback(1,1) = p.buffer(readindex1,1);

% násobení zpětnovazební maticí A
p.feedback = p.feedback(1,:) * p.A;

% součet výstupů linek a násobení výstupním zesílením c
fdn_output = p.buffer(readindex1,1) * p.c1;

% filtr tonální korekce
[wet_signal,p.LPstate] =...
filter(p.LPnum,p.LPden,fdn_output,p.LPstate);

% vytvoření výstupu a dry/wet mix
out(i,:) =...
[in(i,1) * (1-p.wet) + wet_signal * p.wet...
 in(i,2) * (1-p.wet) + wet_signal * p.wet] * p.output_gain;
```

### 3.4 Testování a měření

Testování algoritmu probíhalo v *Matlabu* pomocí nástroje *Audio Test Bench* a jako přeložený VST zásuvný modul v DAW *FL Studio*.

Před spuštěním nástroje *Audio Test Bench* musí algoritmus projít dynamickým testováním na přítomnost chyb v kódu. Příkaz `validateAudioPlugin` spustí nástroj, který postupně posílá na vstup zásuvného modulu bloky audio signálu různé velikosti a testuje, jestli je algoritmus dokáže zpracovat a vrátit v požadované podobě. Například počet řádků na výstupu musí souhlasit s počtem řádků na vstupu. Velikost rámce je na začátku testování 1 vzorek, následně 3 vzorky, dále se zvyšuje na 7, 15, 31 až do 8191.

### 3.4.1 Testování v Audio Test Bench

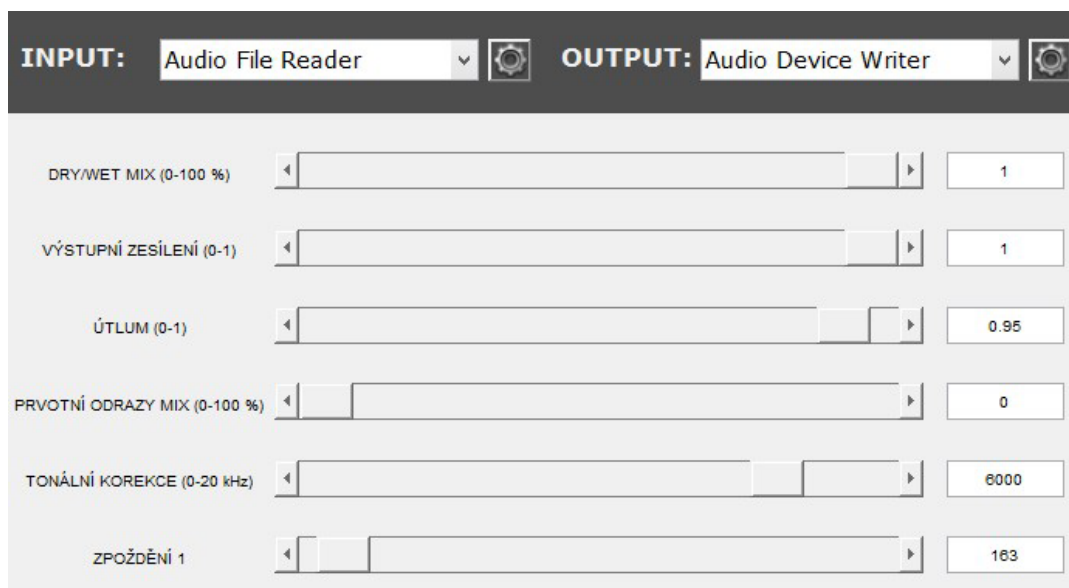
*Audio Test Bench* po spuštění provádí následující operace:

- volání funkce *reset*,
- nastavení veřejných vlastností propojených s parametry ovládacích prvků,
- volání funkce *process* (opakovaně, dokud je na vstupu signál).

*Audio Test Bench* byl používán převážně s nastavením, kdy na vstupu byl 1 sekundu dlouhý zvukový soubor s jednotkovým impulzem. S pomocí nástroje *Time Scope*, který poskytuje vizualizaci signálu v časové doméně, tak bylo možné zobrazit impulzní odezvu reverberátoru. Toto zobrazení se ukázalo jako velmi užitečné pro vývoj algoritmu a sledování vlivu provedených změn v kódu. Stejně tak užitečné bylo toto zobrazení při zjišťování vlivu interaktivních ovládacích prvků na tvar impulzní odezvy. Experimentálně lze najít vhodné kombinace nastavení, např. zpoždění a zesílení vstupních a výstupních větví, pro dosažení požadovaného tvaru impulzní odezvy.

Tímto způsobem bylo například zjištěno, že impulzní odezva dosahuje vysoké hustoty odrazů, pokud je parametr *útlum* nastavený v rozmezí 0,8 až 1,0. Dále bylo zjištěno, že prvotní odrazy sice dokáží vytvořit delší dobu dozvuku a větší hustotu odrazů, ale v impulzní odezvě se objeví opakující se výrazné shluky odrazů v pravidelném intervalu. To způsobuje nežádoucí zabarvení ve výstupním signálu. Maximální dosažitelná doba dozvuku se pohybuje okolo 1 sekundy.

Na obrázku 3.2 je vidět část grafického prostředí nástroje *Audio Test Bench*. Ovládací prvky jednotlivých parametrů jsou rozmístěny pod sebou, a jejich hodnotu lze dynamicky měnit posuvníkem nebo zadáním přesné hodnoty.



Obr. 3.2: Grafické prostředí nástroje *Audio Test Bench*.

*Audio Test Bench* umožňuje přiřadit na vstup libovolný zvukový soubor ve formátu *wav* nebo *ogg* a testovat algoritmus v reálném čase. Tato funkce fungovala pouze při testování implementace jednoduchého filtru a jednoduché mnohonásobné ozvěny. Algoritmus realizující dozvuk ale nedokázal fungovat plynule. Výstup je zkreslený, objevují se neustálé výpadky a praskání. Problém je nejspíše způsobený výpočetní náročností tohoto nástroje a nelze jej vyřešit ani změnou zvukových ovladačů z DirectSound na ASIO.

### 3.4.2 Testování v hostitelské aplikaci

Teprve testování zásuvného modulu v DAW umožnilo zjistit chování algoritmu při souvislém vstupním zvukovém signálu. Díky testování v hostitelské aplikaci lze slyšet projev změn parametrů v reálném čase. Zvuk neobsahuje žádné negativní artefakty způsobené chybami při zpracování, ale samotný charakter dozvuku reverberátoru nedosahuje vysoké kvality. Je slyšet nežádoucí kovový charakter dozvuku, který je silnější při plné intenzitě prvotních odrazů a při některých kombinacích hodnot velikosti zpoždění. Použití Hadamardovy matice ve zpětné vazbě problém nevyřešilo.

Vytížení procesoru se pohybuje kolem 50 % při nastaveném primárním zvukovém ovladači i při nastaveném ovladači zvukové karty ASIO. Automatizace změn hodnot parametrů ovládacích funguje, pokud ale zrovna probíhá změna parametru, objevuje se na výstupu praskání. Při manuálním nastavení ovládacích prvků se tato chyba neobjevuje. Zásuvný modul byl testovaný v 32 i 64 bitové verzi *FL Studio* a v DAW *Reaper*. Z neznámých důvodů *Cubase verze 5* nedokáže otevřít ani načíst jakýkoliv zásuvný modul vygenerovaný *Matlabem*.

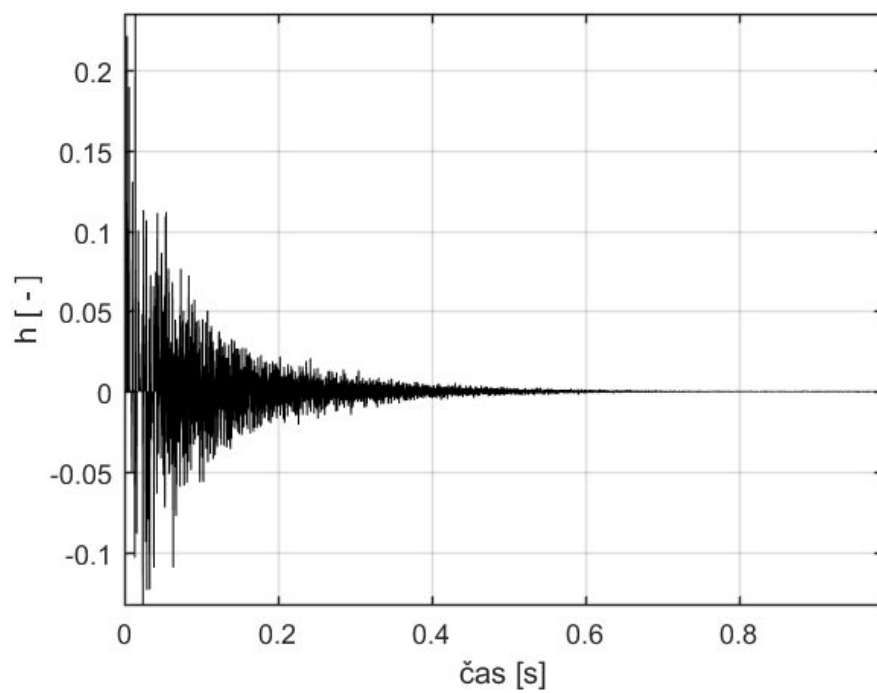
Na obrázku 3.3 je podoba části grafického rozhraní, které *FL Studio* vytváří pro zásuvné moduly bez implementovaného plnohodnotného grafického rozhraní.



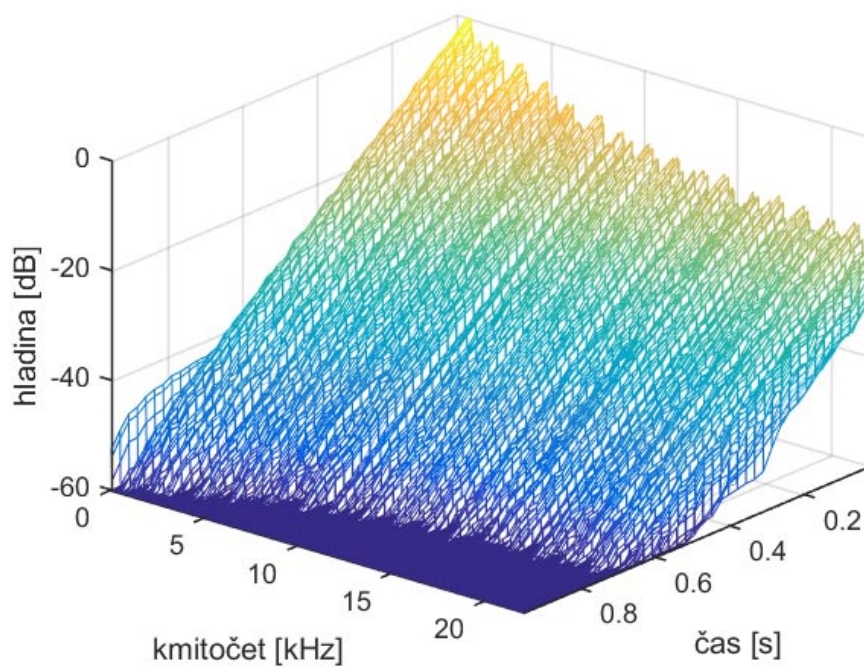
Obr. 3.3: Nativní rozhraní zásuvného modulu v hostitelské aplikaci *FL Studio*.

Tab. 3.1: Objektívni akustické parametry Jotova reverberátoru implementovaného ve VST.

název akustické veličiny	zkratka	jednotka	hodnota
doba dozvuku	$RT$	s	1,00
počáteční doba dozvuku	$EDT$	ms	0,98
míra přímého zvuku	$C_7$	dB	-21,33
jasnost	$C_{50}$	dB	-1,07
míra jasnosti	$C_{80}$	dB	6,82
zřetelnost	$D_{50}$	%	43,85
míra doznívání	$H$	dB	1,07
počáteční prodleva	$ITDG$	ms	3,72
míra hlubokých tónů	$BR$	-	1,01
míra vysokých	$HFR$	-	1,00
těžiškový čas	$T_s$	ms	74,42
kritérium ozvěny	$EK_{max}$	-	0,53



Obr. 3.4: Impulzní odezva Jotova reverberátoru implementovaného ve VST.



Obr. 3.5: Reliéf poklesu energie Jotova reverberátor implementovaného ve VST.

## 4 VÝSLEDKY PRÁCE

### 4.1 Vyhodnocení implementace algoritmů v prostředí Matlab

#### 4.1.1 Dattorův reverberátor

Při implementaci Dattorrova algoritmu je potřeba vzít do úvahy, že se jedná o rekurzivní síť, jak bylo poznamenáno v teoretické části práce. To znamená že na vstupu je potřeba znát předchozí výstupy. To je realizováno tak, že před vstupem signálu je vypočtena přenosová funkce celé struktury *tank*. Výpočet celkové přenosové funkce  $T(z)$  je nejnáročnější operací celého algoritmu [7]. Z pohledu složitosti implementace je nejnáročnějším krokem vytváření výstupů uvnitř struktury fázovacích článků.

Parametry *vstupní difúze 1* a *vstupní difúze 2* mají značný vliv na tvar impulzní odezvy. Ovlivňují zejména výskyt výraznějších shluků odrazů a jejich rozestup. Velký vliv mají také fázovací články na začátku jednotlivých větví struktury *tank*. Pohled na průběh impulzní odezvy na obr. 2.3 odhaluje, že rychlost nárůstu hustoty odrazů není pro kategorii *plate* reverberátorů dostatečná. Naopak samotná hustota odrazů je vysoká v porovnání s ostatními probíranými algoritmy. Reverberátor při poslechovému testu na zvukovém materiálu vykazoval známky tonálního zabarvení a kovový charakter dozvuku. Z reliéfu poklesu energie na obr. 2.4 je patrné, na kterých kmitočtech se objevují rezonanční složky způsobující tonální zabarvení. Tyto složky se stávají výrazné až v doznívací fázi průběhu poklesu energie. Stereofonní výstup vytvořený na základě nastavení různého zpoždění a polarit výstupních bodů podle tabulky 2.1 zvyšuje kvalitu vjemu prostoru.

#### 4.1.2 Gardnerův reverberátor

Implementace Gardnerova algoritmu s konfigurací *Large room* v *Matlabu* není příliš efektivní, z důvodu častého volání funkce *konvoluce* pro získání čísel a jmenovatelů přenosových funkcí jednotlivých funkčních bloků. Komplexnost algoritmu je zvýšena implementací struktur, které se nazývají vnořený fázovací článek. Algoritmus obsahuje jednoduchý vnořený fázovací článek (SNAP) a dvojitý vnořený fázovací článek (DNAP). Modul TDL tvořící prvotní odrazy patří mezi jednodušší způsoby implementace prvotních odrazů. Chyba, která způsobuje nezvykle velkou počáteční prodlevu ITDG, není způsobena chybou při implementaci. Stejný problém zaznamenali při měření vlastností Gardnerova reverberátoru i zde [23]. Tato chyba je vyřešena odstraněním počáteční části odezvy o délce, která odpovídá prodlevě.

Při radikálnějších změnách nastavení reverberátoru však hrozí ořezání počátku impulzní odezvy difúzní části, nebo naopak mezera mezi prvotními odrazy a difúzní částí dozvuku.

U tohoto algoritmu se podařilo dosáhnout nejdelší doby dozvuku. Pro řízení dozukových vlastností Gardnerova algoritmu záleží především na nastavení zesílení a zpoždění pro vnořené fázovací články. Parametr *zpětnovazební zesílení* má z neznámých důvodů jen malý vliv na tvar impulzní odezvy. Tvar výstupní odezvy lze také ovlivnit koeficientem zesílení pro každý ze tří výstupních bodů. Výstup je pouze monofonní. Z EDR na obr. 2.6 je patrné, že výstup vykazuje menší tonální zabarvení, než Dattorův algoritmus. Tento fakt byl potvrzen i poslechovým testem.

### 4.1.3 Jotův reverberátor

Tento algoritmus nevytváří přenosové funkce pro následnou filtraci. Z důvodu omezení prostředí *Matlab* je odezva reverberátoru získána pomocí cyklu `for`. Pro implementaci byla zvolena ortogonální matice, tzn. jedná se o UFDN reverberátor. Při použití Hadamardovy matice vznikaly potíže s nestabilitou a i po nalezení vhodné hodnoty koeficientů matice nebyl výsledek srovnatelný s kvalitou výstupu ortogonální matice. Výstup je vytvořený pomocí konvoluce vytvořené impulzní odezvy se vstupním signálem. Výpočet dozvuku pro jeden vstupní vzorek je téměř okamžitý. Pokud je vstupní signál delší, například několik sekund, trvá výpočet dozvuku mnohonásobně déle.

Pro kvalitu výstupního signálu Jotova algoritmu je rozhodující forma zpětnovazební matice  $\mathbf{A}$ , kde je rozhodující volba typu matice (ortogonální, Hadamardova, Householderova). Důležitá je také volba hodnot koeficientů v matici tak, aby nevyšla nestabilita. Dalším kritickým faktorem, který ovlivňuje vlastnosti dozvuku je nastavení zpoždění zpožďovacích linek. Základní podmínka je zvolit prvočíslo. Neexistuje ale soubor dopředu ověřených hodnot zpoždění, a tak je nutné postupovat empiricky. Při pohledu na reliéf poklesu energie na obr. 2.8 je oproti předchozím algoritmům vidět celková uhlazenost odezvy bez tonálního zbarvení. Přesto při poslechovém testu zpracovaný signál nevykazoval známky absolutní absence tonálního zbarvení. Výstupní filtr tonální korekce filtruje s nedostatečným útlumem v nepropustné oblasti. Nedostatkem je monofonní výstup.



#### 4.1.4 Výsledek porovnání

Každý z těchto algoritmů využívá jiného přístupu k získání výsledku, a tak nelze aplikovat přímé porovnání vlastností. Dattorro navrhoval svůj algoritmus jako reverberátor kategorie *plate*. Gardner zamýšlel svůj reverberátor pro simulaci dozvuku v místnosti. Jot zamýšlel vytvořit co nejuniverzálnější prototyp reverberátoru, který lze použít pro široké spektrum účelů.

Přesto lze vyvodit, že Jotův algoritmus má navzdory neefektivní implementaci v *Matlabu* největší potenciál, protože jeho struktura lze nejsnáze rozšířit. Například přidáním dalších zpožďovacích linek. Nabízí možnosti pro zásadní ovlivnění tvaru výstupní odezvy, na rozdíl od Dattorova a Gardnerova algoritmu, a to zejména volbou zpětnovazební matice. Nedostatek spočívá v hledání nastavení pro dosažení optimálního výsledku empirickou cestou.

Každý z algoritmů obsahuje na konci skriptu příkaz pro spuštění měřicího skriptu, pomocí kterého lze analyzovat objektivní akustické parametry prostoru, impulzní odezvu, křivku a reliéf poklesu zvukové energie pro výstupní signál. Ten lze přehrát a zapsat do souboru ve formátu *wav*. Topologie jsou rozšířené oproti předlohám o větev pro nezpracovaný signál, která se na výstupu sčítá se zpracovaným signálem v poměru zadaném parametrem *dry/wet*. Tímto lze dosáhnout větší kontroly při užití na reálných zvukových nahrávkách, a simulovat tak reálné podmínky použití reverberátoru v hudbě.

Komplexnost těchto algoritmů se dá určit na základě počtu matematických operací potřebných pro jeden blok, pomineme-li fakt, že různé bloky mají různou výpočetní náročnost. Za těchto zjednodušujících podmínek se dá považovat Jotův algoritmus se čtyřmi zpožďovacími linkami za nejefektivnější. Mezi algoritmem Gardnera a Dattorra není v rychlosti zpracování zásadní rozdíl. *Matlab* nabízí možnost změření doby provádění algoritmu pomocí funkcí *tic* a *toc*. Po změření a zprůměrování 4 hodnot pro každý algoritmus vyšly následující hodnoty:

Tab. 4.1: Porovnání doby provádění výpočtu dozvuku pro jeden vzorek.

název algoritmu	doba provádění výpočtu [s]
Dattorro	3,1
Gardner	3,6
Jot	1,5

## 4.2 Vyhodnocení implementace Jotova algoritmu pomocí technologie VST

### 4.2.1 Shrnutí možností rozšíření Audio System Toolbox

Práce s *Audio System Toolbox* při tvorbě VST zásuvného modulu odhalila silné i slabé stránky tohoto rozšíření. Samotná existence možnosti vytvořit VST zásuvný modul přímo v *Matlabu* je vítaná. Při návrhu algoritmů pro zpracování zvuku je důležité znát vlastnosti a chování algoritmu v reálném čase, přestože prvotní implementace probíhá obvykle mimo reálný čas. Pro vytvoření funkčního zásuvného modulu v *AST* stačí uživateli znalost základních pojmů z oblasti OOP. Také je vhodné rozumět principu funkce zásuvných modulů v hostitelských aplikacích.

Transformace kódu algoritmu Jotova reverberátoru do šablony *AST* se ukázala jako neproveditelný krok. Bylo potřeba přehodnotit koncepci zpracování vstupního signálu od samého základu. Testování vytvořeného zásuvného modulu v *Audio Test Bench* otevřelo možnost testovat VST zásuvný modul v reálném čase ještě před jeho vygenerováním do souboru s příponou *dll*. Umožnilo také zjistit impulzní odezvu reverberátoru při nastavení jednotkového impulzu na vstup. Testování na souvislém signálu nebylo možné, protože výstup byl značně porušený. Proto zůstává otázkou efektivita překladačů funkcí do strojového kódu. Vytížení procesoru se při spuštění zásuvného modulu v hostitelské aplikaci pohybuje kolem 50 %. Tato hodnota ukazuje buď na nedostatečně efektivní návrh implementace algoritmu, nebo na nízkou efektivitu přeložení kódu, které provádí *Matlab*.

### 4.2.2 Výsledek implementace pomocí technologie VST

Pro realizaci zpožďovacích linek byla použita struktura, která se nazývá kruhová vyrovnávací paměť. Smyčka `for`, která dříve vytvářela celou impulzní odezvu, nyní přijímá sled bloků vstupního signálu a výstup je tvořen vzorek po vzorku v průběhu jednotlivých iterací smyčky `for`. Funkce probíhající uvnitř smyčky je jádrem celého algoritmu. Ve zpětné vazbě je použita stejná ortogonální matice, jako v původní implementaci. Výchozí hodnoty zpoždění 163, 271, 467 a 829 vzorků při vzorkovacím kmitočtu 44100 Hz byly vybrány empirickou metodou se snahou získat co nejméně zabarvenou odezvu a zároveň co možná největší hustotu odrazů.

Výsledkem implementace do VST je dynamicky připojovaná knihovna s příponou *dll* (*dynamic linked library*), která slouží jako rozšiřující zásuvný modul pro DAW, které má úlohu hostitelské aplikace. VST zásuvný modul v prostředí hostitelské aplikace *FL Studio* pracuje bez vážnějších chyb. Jediná, která byla zaznamenána, se týká výpadků zvuku a praskání při přehrávání a zároveň probíhající změně hodnoty

parametru ovládacího prvku pomocí automatizační křivky. Manuální změna funguje bez problémů v reálném čase.

Zvukový výstup neobsahuje žádné nežádoucí artefakty a pro případ limitace výstupního signálu je součástí ovládacích prvků potenciometr *výstupní zesílení*, který sníží výstupní úroveň zpracované i nezpracované větve signálu. Testováním bylo zjištěno, že není možné použít rozsah zpoždění 1 až 5000 vzorků, protože při hodnotách zpoždění blízkých se 1, vzniká nestabilita systému. Proto je aplikováno ošetření, které nastavuje jako nejmenší hodnotu zpoždění 10 vzorků.

Při simulaci reálného použití reverberátoru na zvukovém signálu, tj. nastavení *dry* = 80 %, *wet* = 20 %, a adekvátním nastavení filtru tonální korekce lze dosáhnout decentně znějícího výsledku. Důvodem je maskování zpracovaného signálu a potlačení kmitočtových složek s nízkou intenzitou v doznívající části dozvuku. Zpracovaný signál na výstupu, tj. *wet* = 100 %, vykazuje známky kovového zabarvení a při poslechu zvukového materiálu, který obsahuje převážně tranzientní složky signálu, se objevují nápadné opakující se zvukové vzorce. Výstupní signál svým zvukovým charakterem připomíná reverberátory s konfigurací *malá místnost*. Tomu odpovídá i to, že reverberátor je nyní schopen vytvořit dozvuk s maximální hodnotou okolo jedné sekundy. Objektívni výsledky jsou zaznamenány jako objektívni akustické parametry a nachází v tab. 3.1. Průběh impulzní odezvy v časové rovině vykazuje požadovaný exponenciálně klesající průběh amplitudové obálky a neobsahuje žádné pravidelně se opakující útvary, viz 3.4. Reliéf poklesu energie na obr. 3.5 obsahuje pravidelné útvary, které jsou zobrazené jako šrafování, a dokazuje, že při zpracování v reálném čase není dostatečný prostor pro vytvoření stejného počtu odrazů, jako při implementaci mimo reálný čas.

Prostorový vjem je znemožněn existencí pouze monofonního výstupu, resp. dva výstupní kanály obsahující identická data. Kompenzaci ztráty této důležité prostorové informace lze v DAW provést přidáním zvukového efektu, který vytváří umělý stereofonní signál. V tomto případě byl použit volně dostupný zásuvný modul s názvem *Dimension Expander*.

### 4.2.3 Možnosti dalšího rozšíření

Rozšířené verze topologie mají 12 až 16 zpožďovacích linek a obsahují ve své struktuře frekvenčně závislé absorpční filtry pro každou linku. Použití absorpčních filtrů bylo dalším pokračováním výzkumu J. Jota. V této práci byly pro výzkumné a testovací účely použity 4 zpožďovací linky a absorpční zesílení definované v rovnici 1.7. Jednak z důvodu menších výpočetních nároků a také kvůli přehlednosti vygenerovaného grafického prostředí. Rozložení ovládacích prvků je určené pouze pořadím v jakém jsou seřazeny parametry objektu `PluginInterface`. Při použití například

12 zpožďovacích linek by bylo pro každou z nich potřeba vytvořit 3 ovládací prvky, což ve výsledku včetně zahrnutí nastavení útlumu, filtru tonální korekce a podobně, čítá přes 40 ovládacích prvků v jediném okně.

Pokud bude uvažováno rozšíření v rámci vývojového prostředí *Matlab*, lze se zaměřit na vylepšení dozvukových vlastností reverberátoru použitím absorpčních filtrů pro každou zpožďovací linku [22]. Při zvýšení počtu zpožďovacích linek je nutné zvolit novou zpětnovazební matici, která bude stále odpovídat podmínce 1.6. Zůstává otázkou, zda bude výsledek komplexnější implementace fungovat bez obtíží a s přijatelnou mírou latence.

Při uvažování nad případným rozšířením a dalším pokračováním práce je na místě připomenout, že *Audio System Toolbox* je navržený především jako mezičlánek při vývoji VST zásuvných modulů. Po implementaci algoritmu zpracovávajícího data mimo reálný čas, jeho úpravě pro zpracování v reálném čase a vygenerování testovacího VST zásuvného modulu, je dalším krokem finální implementace algoritmu v jazyce *C++*.

## ZÁVĚR

Cílem práce byla implementace algoritmů syntetických reverberátorů, které navrhli Jon Dattorro, William Gardner a Jean-Marc Jot v prostředí *Matlab*, a následné porovnání možností řízení dozvukových vlastností a komplexnosti algoritmů. Dále je Jotův algoritmus implementován pomocí technologie VST v prostředí *Matlab* s využitím rozšíření, které se nazývá *Audio System Toolbox*.

První část práce obsahuje teoretické poznatky k jednotlivým reverberátorům, v druhé části je popsán způsob implementace algoritmů v *Matlabu*. Třetí část specifikuje vlastnosti rozšíření *Audio System Toolbox* a následuje popis implementace Jotova algoritmu pomocí technologie VST. Další část obsahuje porovnání a zhodnocení dosažených výsledků při implementaci v *Matlabu* s ohledem na možnosti řízení dozvukových vlastností a složitost algoritmů. V druhé polovině této části je věnován prostor pro zhodnocení výsledků implementace pomocí technologie VST a naznačeny možnosti případného dalšího pokračování této práce. Jsou zde také zhodnoceny vlastnosti vývojového prostředí.

Výsledným produktem této práce jsou skripty v *Matlabu*, ve kterých jsou implementovány tři různé algoritmy pro syntetické modelování dozvukových vlastností místnosti. Změnou vstupních parametrů lze měnit vlastnosti vygenerovaného dozvuku. Pomocí přiloženého souboru s názvem *measure tool*, který provádí měření, lze změřit a zaznamenat objektivní akustické parametry prostoru, které slouží i pro popis vlastností reverberátorů. Je možné vygenerovat průběh impulzní odezvy v časové rovině, zobrazit křivku poklesu energie, a také reliéf poklesu energie. V rámci testování je jako vstupní signál nastavený jednotkový impulz uložený ve zvukovém souboru. Na vstup lze přivést libovolný signál, ovšem výsledky měření budou poté irelevantní.

Dalším produktem je Jotův reverberátor v podobě VST zásuvného modulu a skript s jeho implementací. Ten může sloužit jako prostředek ke zkoumání vlastností reverberátoru založeného na unitární zpětnovazební zpožďovací síti, který pracuje v reálném čase. Vlastnosti zjištěné z impulzní odezvy jsou porovnány s impulzní odezvou skutečné místnosti.

# LITERATURA

- [1] SCHRÖEDER, M. R. a B. F. LOGAN. *Colorless Artificial Reverberation*. Murray Hill, New Jersey: Bell Telephone Laboratories, Incorporated, 1961.
- [2] MOORE, F. Richard. *Elements of Computer Music*. 1. Englewood Cliffs, NJ: Prentice-Hall, 1990. ISBN 978-0132525527.
- [3] SCHRÖEDER, M. R. *Natural sounding artificial reverberation*. J. Audio Eng. Soc. Murray Hill, New Jersey: Bell Telephone Laboratories, Incorporated, 1962, 10(3), strana 219-233.
- [4] JOT, Jean-Marc. *Efficient models for reverberation and distance rendering in computer music and virtual audio reality*. 1 place Igor-Stravinsky. F-75004 Paris, France: IRCAM, 1992.
- [5] BALÍK, Miroslav. *Simulace akustických prostorů - modely pro simulace v reálném čase* [online]. [cit. 2017-05-28]. Dostupné z: <<http://www.elektrorevue.cz/clanky/01016/index.html>>.
- [6] DATTORRO, Jon. *Effect Design. Part 1: Reverberator and Other Filters*. Journal of Audio Engineering Society. Stanford, CA, USA: CCRMA, Stanford University, 1997, 45(9). ISSN 660-684.
- [7] BELTRÁN, F. A., J. R. BELTRÁN, N. HOLZEM a A. GOGU. *Matlab Implementation of Reverberation Algorithms*. Journal of New Music Research. 2002, 31(2), strana 156-161. ISSN 0929-8215.
- [8] SMITH, Julius O. *Physical Audio Signal Processing for Virtual Musical Instruments and Digital Audio Effects: FDN Reverberation* [online]. 2010 [cit. 2017-05-15]. ISBN 978-0-9745607-2-4. Dostupné z: <[https://ccrma.stanford.edu/~jos/pasp/FDN\\_Reverberation.html](https://ccrma.stanford.edu/~jos/pasp/FDN_Reverberation.html)>.
- [9] VERCOE, B., MILLER a PUCKETTE. *Synthetic Spaces – Artificial Acoustic Ambience from Active Boundary Computation*. 1985.
- [10] GARDNER, W. G. *The Virtual Acoustic Room*. Cambridge, Massachusetts, 1992. Diplomová práce. Computer Science and Engineering Massachusetts Institute of Technology,.
- [11] JOT, Jean-Marc. *Etude et réalisation d'un spatialisateur de sons par modèles physiques et perceptifs*. Paris, 1992. Disertační práce. Telecom Paris.

- [12] KAHRS, Mark a Karlheinz BRANDENBURG. *Applications of Digital Signal Processing to Audio and Acoustics*. 1. USA: Kluwer Academic Publishers, 2002. ISBN 0-7923-8130-0.
- [13] STAUTNER, J. a M. Puckette. *Designing multichannel reverberators*. Computer Music Journal, 6(1), strana 52–65, 1982.
- [14] GOGU, Adrian a Marina TOPA. *Coefficients' Computation for Jot's Reverberation Algorithm*. 10th Mediterranean Electrotechnical Conference, MEleCon. IEEE, 2000, 10(1), strana 45-48. ISSN 7803-6290.
- [15] COGGIN, Jay a Will PIRKLE. *Automatic Design of Feedback Delay Network Reverb Parameters for Impulse Response Matching*. AES 141. Convention. Los Angeles, USA, 2016.
- [16] HAAS, H. *Über den Einfluss eines Einfachechos auf die Hörsamkeit von Sprache*. Acustica. 1951.
- [17] VONDRÁŠEK, M. a M. ANTEK ML. *Porovnání objektivních kritérií kvality koncertních sálů*. Akustické listy. 2005, 11(3), strana 9-18.
- [18] *Audio System Toolbox: Getting Started Guide (r2017a)* [online]. 3. Natick, Massachusetts: Mathworks, Inc, 2017 [cit. 2017-05-15]. Dostupné z: <[https://www.mathworks.com/help/pdf\\_doc/audio/audio\\_gs.pdf](https://www.mathworks.com/help/pdf_doc/audio/audio_gs.pdf)>.
- [19] PIRKLE, Will. *Designing Audio Effect Plug-Ins in C++: With Digital Audio Signal Processing Theory*. 1. 70 Blanchard Road, Suite 402, Burlington, MA 01803, USA: Focal Press, 2013. ISBN 978-0-240-82515-1.
- [20] *Audio System Toolbox: Reference Guide (r2017a)* [online]. 3. Natick, Massachusetts: Mathworks, Inc, 2017 [cit. 2017-05-15]. Dostupné z: <[https://www.mathworks.com/help/pdf\\_doc/audio/audio\\_ref.pdf](https://www.mathworks.com/help/pdf_doc/audio/audio_ref.pdf)>.
- [21] DEVANE, Charlie a Gabriele BUNKHEILA. *Automatically Generating VST Plugins from MATLAB Code*. AES Convention e-Brief 238 [online]. Natick, MA, USA: Mathworks, 2016 [cit. 2017-05-16]. Dostupné z: <[https://www.mathworks.com/tagteam/89231\\_PluginGenerationEbriefMWFormat.pdf](https://www.mathworks.com/tagteam/89231_PluginGenerationEbriefMWFormat.pdf)>.
- [22] JOT, Jean-Marc a Antoine CHAIGNE. *Digital Delay Networks for Designing Artificial Reverberators*. AES 90. Convention. 1991.
- [23] TOMA, Norbert, M. D. TOPA, V. POPESCU a E. SYOPOS. *Comparative Performance Analysis of Artificial Reverberation Algorithms*. Str. G. Barițiu

nr. 26-28, 400027 Cluj-Napoca, Romania: Technical University of Cluj-Napoca, 2005.



## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AGU	jednotka pro generování adres v DSP procesorech – Address Generation Unit
AP	fázovací členek – All Pass Filter
API	grafické rozhraní aplikace – Application Program Interface
ASIO	ovladač zvukové karty pro správu vstupů a výstupů s nízkou latencí – Audio Stream Input/Output
AST	rozšiřující software Matlabu – Audio System Toolbox
BR	poměr hlubokých tónů – Bass ratio
DAW	počítačový software pro nahrávání, editaci a produkci zvukových souborů – Digital Audio Workstation
DNAP	dvojitý vnořený fázovací členek – Double Nested All Pass Filter
DP	filtr typu dolní propust
DSP	digitální zpracování signálu – Digital Signal Processing
EDT	počáteční doba dozvuku – Early Decay Time
FIR	diskrétní systém s konečnou impulzní charakteristikou – Finite Impulse Response
FDN	zpětnovazební zpožďovací síť – Feedback Delay Network
HFR	poměr vysokých tónů – High Frequency Ratio
IIR	diskrétní systém s nekonečnou impulzní charakteristikou – Infinite Impulse Response
ITDG	počáteční prodleva – Initial Time Delay Gap
LP	filtr typu dolní propust – Low Pass Filter
LTI	časově invariantní systém definovaný impulzní odezvou – Linear Time Invariant
MIDI	standard pro komunikaci mezi hudebními nástroji a počítačem – Musical Instrument Digital Interface

OOP	objektově orientované programování – Object Oriented Programming
RT	doba dozvuku – Reverb Time
SNAP	jednoduchý vnořený fázovací článek – Single Nested All Pass Filter
TDL	zpožďovací linka s přidávanými výstupy uvnitř struktury – Tapped Delay Line
VST	technologie zásuvných modulů – Virtual Studio Technology
UFDN	zpětnovazební zpožďovací síť s unitární maticí ve zpětnovazební smyčce – Unitary Feedback Delay Network

# SEZNAM PŘÍLOH

A	Vstupní parametry reverberátorů	67
B	Obsah přiloženého DVD	70

## A VSTUPNÍ PARAMETRY REVERBERÁTORŮ

Tab. A.1: Vstupní parametry Dattorrova reverberátoru.

<b>název parametru</b>	<b>hodnota</b>
mezní kmitočet filtru typu DP	2500
vstupní difúze 1	0,95
vstupní difúze 2	0,625
difúze poklesu 1	0,7
difúze poklesu 2	0,5
difúze poklesu 3	0,5
útlum	0,0005
pokles	0,5
fázovací článek 1	142
fázovací článek 2	107
fázovací článek 3	379
fázovací článek 4	277
fázovací článek 5.1	672
fázovací článek 6.1	1800
zpoždění 1.1	4453
zpoždění 1.2	3720
fázovací článek 5.2	908
fázovací článek 6.2	2656
zpoždění 2.1	4217
zpoždění 2.2	3163

Tab. A.2: Vstupní parametry Gardnerova reverberátoru.

<b>název parametru</b>	<b>hodnota</b>
mezní kmitočet	2600
zesílení zpětnovazební větve	0,4
zesílení fázovacího článku 1	0,3
zesílení fázovacího článku 2	0,3
zesílení vnějšího fázovacího článku	0,5
zesílení vnitřního fázovacího článku 1	0,25
zesílení vnitřního fázovacího článku 2	0,25
zesílení výstupu 1	0,34
zesílení výstupu 2	0,14
zpoždění fázovacího článku 1	352
zpoždění fázovacího článku 2	529
zpoždění 1	176
zpoždění 2	749
zpoždění 3	1367
zpoždění 4	661
Zpoždění vnějšího vnořeného fázovacího článku	3826
Zpoždění vnitřního vnořeného fázovacího článku	3734
Zpoždění vnějšího dvojitého vnořeného fázovacího článku	5292
Zpoždění vnitřního dvojitého vnořeného fázovacího článku 1	3351
Zpoždění vnitřního dvojitého vnořeného fázovacího článku 2	1323

Tab. A.3: Vstupní parametry Jotova reverberátoru.

<b>název parametru</b>	<b>hodnota</b>
mezní kmitočet	5000
doba dozvuku	1,5
zpoždění 1	1847
zpoždění 2	1979
zpoždění 3	2281
zpoždění 4	2557
zesílení zpětnovazební větve $b_1$	1
zesílení zpětnovazební větve $b_2$	1
zesílení zpětnovazební větve $b_3$	1
zesílení zpětnovazební větve $b_4$	1
výstupní zesílení $c_1$	1
výstupní zesílení $c_2$	1
výstupní zesílení $c_3$	1
výstupní zesílení $c_4$	1

## B OBSAH PŘILOŽENÉHO DVD

- elektronická verze Bakalářské práce
- návod a informace ke spuštění skriptů a VST zásuvného modulu
- složka obsahující skripty Dattorrova algoritmu
- složka obsahující skripty Gardnerova algoritmu
- složka obsahující skripty Jotova algoritmu
- složka obsahující skripty Jotova algoritmu implementovaného pomocí technologie VST
- VST zásuvný modul `jot_reverb_vst.dll`